Steered Mixture-of-Experts for Image and Light Field Representation, Processing, and Coding:

A Universal Approach for Immersive Experiences of Camera-Captured Scenes

vorgelegt von
M. Sc.
Ruben Verhack
ORCID: 0000-0001-6636-629X

an der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Doktorväter:	Prof. Dr. Peter Lambert
	Prof. Dr.-Ing. Thomas Sikora
Vorsitzende:	Prof. Dr. Ir. Gert De Cooman
	Prof. Dr.-Ing. Rafael Schaefer
Gutachter:	Prof. Dr. Ir. Glenn Van Wallendael
	Prof. Dr. Klaus Obermayer
	Prof. Dr. Ir. Christine Guillemot
	Dr. Ir. Jean-François Macq
	Dr. Ir. Tim Wauters

Tag der wissenschaftlichen Aussprache: 28. April 2020

Berlin 2020

Steered Mixture-of-Experts' voor de representatie, verwerking en codering van beelden en lichtvelden: een universele aanpak voor immersieve belevingen op basis van camerabeelden

Steered Mixture-of-Experts for Image and Light Field Representation, Processing and Coding: A Universal Approach for Immersive Experiences of Camera-Captured Scenes

Ruben Verhack

Promotors: Prof P. Lambert, PhD, Prof T. Sikora, PhD
Doctoral thesis submitted in order to obtain the academic degrees of
Doctor of Computer Science Engineering (Ghent University) and
Doktor der Ingenieurwissenschaften (Technische Universität Berlin)

GHENT
UNIVERSITY

Department of Electronics and Information Systems
Head of Department: Prof K. De Bosschere, PhD
Faculty of Engineering and Architecture

Institute of Telecommunication Systems
Head of Institute: Prof T. Sikora, PhD
Faculty of Electrical Engineering and Computer Science

Academic year 2019 - 2020

ISBN 978-94-6355-367-4
NUR 958
Wettelijk depot: D/2020/10.500/44

Promotors:	Prof. Dr. Peter Lambert	UGent, BE
	Prof. Dr.-Ing. Thomas Sikora	TU Berlin, DE
Chairs:	Prof. Dr. Ir. Gert De Cooman	UGent, BE
	Prof. Dr.-Ing. Rafael Schaefer	TU Berlin, DE
Reading Committee:	Prof. Dr. Ir. Glenn Van Wallendael	UGent, BE
	Prof. Dr. Klaus Obermayer	TU Berlin, DE
	Prof. Dr. Ir. Christine Guillemot	INRIA, FR
	Dr. Ir. Jean-François Macq	Nokia Bell Labs, BE
	Dr. Ir. Tim Wauters	UGent, BE
Secretary:	Prof. Dr. Ir. Glenn Van Wallendael	UGent, BE

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Elektronica en Informatiesystemen
IDLab

Campus Ardoyen
Technologiepark-Zwijnaarde 126,
B-9052 Gent, België

Technische Universität Berlin
Fakultät IV Elektrotechnik und Informatik

Fachgebiet Nachrichtenübertragung

Sekr. MAR 6-1,
Marchstraße 23,
D-10587 Berlin, Germany

Acknowledgments

Figure 1: My sister, my father, and myself in 2012 in front of the Haus der Elektrotechnik und Informatik (TU Berlin).

Where to start? It's been so long that I can hardly remember. Still, there I was during my spring semester abroad in 2012 at the TU Berlin. Never could I have thought that that semester would be the start of my journey that eventually led to this work, eight years later. I would truly like to thank Prof. De Turck for motivating me to take on an adventure abroad. Without that little push, none of this would have happened. Similarly, I would like to thank Prof. Van de Walle to have guided me for many years during my student years, as a lecturer, as a mentor during my student-entrepreneurship, and as promotor for my master's thesis abroad. Furthermore, he was one of the professors that for the first time really made me believe that you can dream big even coming from a small city in a small country.

In Berlin, I finished my master's thesis and continued to work a semester as a scientific researcher under the supervision of Prof. Thomas Sikora. In 2013, I officially started my PhD at TU Berlin. I can not stress enough how much I have learned academically, professionally and personally under his supervision. It was an honor to have worked for someone as esteemed and experienced. During my PhD, he was extremely generous to share his international network and he never skipped an opportunity to promote our work wherever he was. The biggest lesson was to never underestimate your own work and not to be afraid of going against the mainstream technologies.

In 2014, we decided to proceed with my PhD as a collaboration between TU Berlin and UGent. I moved back to Ghent to join the Multimedia Lab where I stayed through multiple mergers and reorganizations (first: Data Science Lab - iMinds, now: IDLab - imec). The dynamics of a lab being led by a young team of professors was very complementary to the experience I had so far. I would like to thank Prof. Jan De Cock and Prof. Peter Lambert for all my years there. Especially, I would like to thank Prof. Peter Lambert for his enduring support in this PhD marathon up to the very end. He helped me through some of the hardest years and always magically found funding for our research. Similarly, I would like to thank Prof. Glenn van Wallendael. As a colleague and as a supervisor, he always found time for a brain storm and a good laugh. I would finally like to thank Prof. Nilesh Madhu for supporting me with his mathematical knowledge and his friendliness.

During this collaboration, I was lucky enough to enjoy the best of both worlds by traveling back and forth between Ghent and Berlin every couple of months. I could have never performed this work without the support of my colleagues at both TU Berlin and Ghent University. I was lucky to be in two great work atmospheres. First, at TU Berlin, I would like to thank Lieven Lange, Rolf Jongebloed, Michael Tok and Erik Bochinski for all the in-depth discussions, support, and the many after-work beers at Café Shila or at the Spree. At UGent, there are many people that I would like to thank. Especially I would like to thank the Multimedia Lab veterans, Niels Van Kets, Vasileios Avramelos, Johan De Praeter, Tom Paridaens and Ignace Saenen. Additionally, I would also like to thank the new cadets, Martijn Courteaux and Hannes Mareen. Especially Martijn, as he showed that a supervisor can learn a lot from his student. Finally, there is one person who is the ultimate veteran. He stood by my side since day one at university up until we both miraculously finished our PhDs in the same academic year at the same lab. This person is Baptist Vandersmissen, and life would have been plain boring without him and I guess he's smart too.

Of course, life is more than just work. For that reason, I would definitely like to thank my best friends Wout and Michiel for all their support and good times over all those years. Similarly, my family always stood by my side. I would like to thank my brother Lander, my sister Friedel and Mieke. However, I would especially like to thank my father, Luc Verhack, who went through extreme lengths to provide me with all the opportunities that I enjoyed, to support me financially throughout university, and to believe that "a guy like me should be able to get better grades". Of course, I would like to thank my mother, Anny Deschildre, who was taken from us decades too early. And finally, I would like to thank my girlfriend Lily Wakelin for all the love, the laughs, the great food, and to have supported me emotionally through the hardest part of my PhD. Of course, there are many people that I crossed paths with over these eight years. I wished I could thank them all.

Gent, April 2020
Ruben Verhack

Selbstständigkeitserklärung
–Declaration of Independent Work–

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Ich erkläre, dass mir die geltende Promotionsordnung der TU Berlin bekannt ist. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Berlin, den 13.09.2019
Ruben Verhack

Table of Contents

Acknowledgments **i**

Selbstständigkeitserklärung **iii**

Summary **xi**

Nederlandse samenvatting **xvii**

Deutsche Zusammenfassung **xxiii**

1 Introduction **1**
1.1 Context . . . 1
1.2 Visual Immersive Experiences . . . 3
1.3 Towards 6-Degrees-of-Freedom CC-VR . . . 5
1.3.1 3-D reconstruction . . . 5
1.3.2 Based on video coding . . . 7
1.3.3 The scene-representation continuum . . . 8
1.4 The Proposed Method . . . 9
1.4.1 Compression efficiency . . . 11
1.4.2 Continuous representation . . . 12
1.4.3 Random access . . . 13
1.4.4 Descriptive model . . . 13
1.4.5 Pixel-parallel and light-weight decoding . . . 14
1.5 Conclusion . . . 15
1.6 Outline . . . 16
1.7 Publications . . . 16
1.7.1 Journal publications . . . 16
1.7.2 Conference proceedings . . . 17
1.7.3 Awards . . . 18

2 The Plenoptic Function and Light Fields **19**
2.1 Introduction . . . 19
2.2 The Mathematics of Light . . . 20
2.2.1 The 5-D Plenoptic Function . . . 20
2.2.2 The 4-D Light Field . . . 21

2.2.3 Viewport rendering . . . 23
2.2.4 Images and video as functions . . . 25
2.2.5 What to use for CC-VR? . . . 26
2.3 Current Light Field Processing . . . 27
2.3.1 Light field acquisition . . . 27
2.3.2 Light field super-resolution . . . 30
2.3.3 Light field depth estimation . . . 31
2.3.4 Light field compression . . . 31
2.4 Conclusion . . . 34

3 Steered Mixture-of-Experts **35**
3.1 Introduction . . . 35
3.2 Motivation . . . 36
3.2.1 Related models in the literature . . . 39
3.3 Steered Mixture-of-Experts . . . 41
3.3.1 Mixture-of-Experts . . . 42
3.3.2 Training of Mixture Models with Distributions of the Exponential Family . . . 44
3.3.3 Mixture-of-Experts based on GMMs . . . 45
3.3.4 Example: 1-D Steered Mixture-of-Experts (SMoE) . . . 46
3.3.5 Student-t Mixture Models . . . 47
3.4 Insights into the SMoE Image Model . . . 49
3.4.1 SMoE for sparse image representation . . . 49
3.4.1.1 Example: binary image . . . 50
3.4.1.2 Example: natural image . . . 51
3.4.2 Image descriptors . . . 52
3.4.3 Color representation . . . 53
3.4.4 Resampling, pixel-parallel reconstruction and random access 55
3.5 Image Experiments . . . 55
3.5.1 Dataset . . . 56
3.5.2 GMM vs STM . . . 56
3.5.3 Mean/median/mode estimators . . . 56
3.5.4 Chroma reconstruction . . . 59
3.6 Conclusion . . . 60

4 SMoE for Immersive Image Modalities **63**
4.1 Introduction . . . 63
4.2 SMoE for Spherical Coordinate Dimensions . . . 64
4.2.1 SMoE on the Unit Sphere . . . 65
4.2.2 Projection onto the tangent space . . . 65
4.2.3 Dimensionality reduction . . . 67
4.2.4 Omnidirectional images . . . 68
4.2.5 Conclusion . . . 69
4.3 SMoE for Light Fields . . . 71
4.3.1 View interpolation . . . 72

4.3.2 Depth estimation . . . 75
4.3.3 Edge detection and other descriptors . . . 76
4.3.4 Pixel-parallel real-time view reconstruction . . . 77
4.3.5 SMoE in a light field processing pipeline . . . 79
4.4 SMoE for Light Field Video . . . 79
4.5 Conclusion . . . 83

5 Building Mixture Models From Extremely Large Datasets 85
5.1 Introduction . . . 85
5.2 Insights and Complexity of EM . . . 86
5.2.1 The EM algorithm and latent variables . . . 86
5.2.2 Complexity . . . 88
5.2.2.1 Faster convergence . . . 88
5.2.2.2 Fewer samples per iteration . . . 88
5.2.2.3 Fewer kernels per iteration . . . 89
5.2.2.4 Parallel and hardware implementations . . . 89
5.3 Methods used in SMoE . . . 90
5.3.1 Initialization . . . 90
5.3.2 Regularization . . . 91
5.3.3 Block-based modeling . . . 91
5.3.4 Minibatch EM . . . 92
5.3.4.1 Batch vs. Minibatch experiment . . . 93
5.3.5 Split-and-Merge EM . . . 95
5.3.6 Progressive modeling . . . 96
5.3.6.1 Block-level updates . . . 96
5.3.6.2 Kernel splitting . . . 97
5.3.6.3 Example: modeling light field video . . . 98
5.3.7 Modeling spherical image dimensions . . . 100
5.4 Conclusion . . . 101

6 SMoE for Compression and Coding 103
6.1 Introduction . . . 103
6.2 Relation to FTV and ray-space representation . . . 104
6.3 General Proposed Coding Structure . . . 107
6.3.1 Kernel centers . . . 107
6.3.2 Kernel covariance matrices . . . 107
6.3.3 Entropy coding . . . 108
6.3.4 Coding summary . . . 109
6.4 Image Coding . . . 109
6.4.1 Spatial Activity Analysis and Modeling . . . 110
6.4.2 Difference coding and quantization . . . 111
6.4.3 Entropy coding . . . 111
6.4.4 Image coding experiments . . . 112
6.5 Light Field Coding . . . 115
6.5.1 Window $R_{XX,j}$ quantization . . . 115

6.5.2 Center and slope quantization and arithmetic coding . . . 120
6.5.3 Light field coding experiments . . . 121
6.5.4 Subjective light field experiments . . . 126
6.5.5 Light field view consistency experiments . . . 127
6.6 Light Field Video Coding . . . 128
6.6.1 Light field video coding experiment . . . 130
6.7 Conclusion . . . 133

7 Conclusion and Future Work **137**
7.1 Conclusion . . . 137
7.2 Future work . . . 138

Summary

The introduction of analog photography in the 1800s enabled people to capture their surroundings on film for the first time. Later, photography gave rise to analog cinema by capturing and displaying photographs in a fast-moving sequence. Nowadays, images and video mainly reside in the digital world and have never been as ubiquitous. The digitization of imaging further enabled digital manipulation of the images, ranging from photo editing to incorporating huge amounts of computer-generated imagery (CGI). Over the last decades, computer graphics have made enormous progress in terms of content-creation tools, rendering technology and hardware acceleration. These tools are not only applicable for adding special effects to films, but these techniques also enable us to create entire digital worlds. For example, gamers can roam across large areas and view objects far away or nearby with breathtaking quality. Even better, they can interact with the objects! These concepts would have been incredibly hard to imagine even fifty years ago.

Interestingly, the digitization of imaging and cinema did not change the way in which camera-captured visual data is experienced by the viewer. A movie director captures a specific viewpoint after which the viewers are shown what the director wanted them to see. Now, imagine stepping into the story yourself with photorealistic quality. This would enable you to move around and to explore the space behind objects. Major events could then be captured in such a way that the viewer can relive the event from any point of view as a silent free-roaming spectator. In such a case, viewers can enjoy the navigational freedom as in gaming, but with the photorealism of cinema. Applications are not limited to entertainment or creating "living memories". It is easy to imagine applications in education, cultural heritage preservation, therapy, remote control in industry, medicine, tourism, etc.

Nevertheless, virtual reality (VR) for camera-captured scenes is fundamentally different and more complex compared to VR consumption of computer-generated (CG) scenes (e.g. as in gaming). The problem is much more challenging due to the lack of geometrical knowledge. Currently, camera-captured virtual reality (CC-VR) is mainly limited to 360-degree video which is inadequate at delivering a full VR experience as it does not deliver positional freedom to the user. Viewers can only look around but their head's location is fixed. As such, there is a large interest in overcoming these limitations and allowing the viewer to walk around in a scene over a large spatial extent in between objects and even through doors.

In this dissertation, I present a novel methodology for representing and coding of a wide range of image modalities (e.g. images, video, light fields) that is designed in a future-proof manner to allow for future CC-VR. First, traditional

methods are discussed that are currently pivoted towards CC-VR and requirements are identified for a VR-ready image representation. Secondly, the novel representation is introduced and illustrated for lower-dimensional images (e.g. images) and are then further applied on more immersive image modalities (e.g. 360-degree images and light fields). Finally, a coding method is presented and evaluated in order to efficiently store, transmit and broadcast camera-captured scenes.

At the moment, two general strategies are being followed for allowing true immersive experiences of camera-captured scenes. The first strategy consists of constructing a 3-D model of the scene. However, the geometry and texture approximation steps are inherently lossy, time-consuming and often require manual intervention when the quality is required to be photorealistic. Furthermore, they struggle with non-rigid objects such as smoke, fire, water, and transparent surfaces. The second strategy relies on known video-coding methods. Following this philosophy, scenes are represented by coding a number of camera viewpoints as video, and further reconstructing the missing ones by view synthesis. Video-coding methods provide excellent coding performance for video and for problems that can be translated to video. However, in the long run, it can be argued that this problem-translation becomes less evident for VR with extensive user autonomy. In video coding, videos are represented as a sequence of still images. These images (or "frames") thus have an explicit order, i.e. time. This logical ordering was the greatest source for reducing storage space for videos. Two consecutive frames tend to be very similar. Once one frame is known, then only the difference between that frame and the next frame needs to be stored or transmitted. However, these paradigms are challenged once the viewer is allowed to choose where to look at any time. In other words, the order of the frames is not known at encoding-time but is only decided at playback by the user's movement. Furthermore, the number of possible views grows exponentially with the level of user's autonomy. Moreover, the required view synthesis process to generate the missing views has limitations (e.g. in handling reflections and occlusions).

The proposed *Steered Mixture-of-Experts* (SMoE) methodology models the light information as higher-dimensional data. The reason is that, in essence, the observed 2-D views in a VR scene at each position and gaze orientation are 2-D slices of higher-dimensional light data. The idea of the proposed strategy is to treat higher-dimensional data as such, and to not reduce the data to 2-D sampling grids or to construct the 3-D geometry of a scene. More specifically, the underlying pixel-generating plenoptic function is approximated by using inherently higher-dimensional atoms called 'kernels'. A kernel can be viewed as a multi-dimensional generalization of the pixel. These kernels allow for simultaneous harvesting of pixel color correlation in various directions: e.g. time, pixel position, camera position. Furthermore, a single kernel describes a multi-dimensional gradient along these dimensions.

The benefits of the proposed SMoE representation are as follows. First, the method is scalable in dimensionality and thus can be applied to any-dimensional image modalities. Secondly, a single kernel can cover a large number of originally captured pixels while keeping the number of parameters per kernel limited. There-

fore, the representation is suited for the application of compression. Thirdly, the SMoE model is continuous, therefore, reconstructing a view boils down to merely sampling the model at the desired resolution. Fourthly, the reconstruction of 2-D views based on the model can be performed in a light-weight and pixel-level parallel manner. Fifthly, kernels are not interdependent, in order to reconstruct a portion of the image modality, only the kernels in the vicinity of that part of the coordinate space need to be present. This allows for fine-grained random access when streaming natural VR scenes. Finally, the representation takes on the structure of the data itself. As such, the parameters of the model reveal relevant machine-readable information about the scene (e.g. edges, intensity flow, motion flow, ...) and can even implicitly indicate the geometry of a scene in the case of light fields. Furthermore, the model inherently provides a multi-dimensional segmentation of the image modality.

The SMoE method is based on the data-adaptive division of the coordinate space of the image modality. For any image modality, the coordinate space is defined as $\mathbb{R}^p$ and the color space as $\mathbb{R}^q$. For images, video, light field images, and light field video, the dimensionality p is respectively 2, 3, 4, and 5. For monochrome images, q is 1, and for color images q is typically 3. The goal is to divide the coordinate space $\mathbb{R}^p$ into stationary regions, and to find regressors ($f : \mathbb{R}^p \mapsto \mathbb{R}^q$) that locally approximate this stationary region well. The underlying assumption is that image pixels are instantiations of a non-linear or non-stationary random process that can be modeled by spatially-piecewise stationary stochastic processes. As such, the model takes into account different regions of the image and their segmentation borders. Furthermore, in light fields, the epipolar-plane images consist of diagonally-structured lines, and in video, motion can be approximated by line segments as is done in motion compensation in traditional video coding. Therefore, a piecewise approximation of image modalities is pursued in this dissertation. Each such stationary region is then ideally represented by a single kernel. Such a division of the input space is a general *Mixture-of-Experts* (MoE) strategy, well known in the machine learning field. However, SMoE is based on a mixture model (or "alternative") version of the MoE approach. In this version, both segmentation and local regressors (the *experts*) are derived from the $(p+q)$-dimensional components of a mixture model. This mixture model models the joint probability distribution of a sample coordinate vector $X \in \mathbb{R}^p$ and a sample color vector $Y \in \mathbb{R}^q$. For grayscale images, the model is thus 3-D (two spatial coordinates, one color value), whereas for color light fields the model is 7-D (four coordinates, three color values).

In this dissertation, the focus lies on the *Gaussian Mixture Model* (GMM) case. One Gaussian kernel in the model defines a linear regressor through the conditional $Y|X$, and all kernels combined define a segmentation of the coordinate space. The model thus only consists of a set of Gaussian kernels which are defined by their centers and covariances. The reason for choosing GMMs is that they offer elegant mathematics and limited parametrization. The MoE based on GMMs results in smoothed piecewise approximations. In order to model an image modality, the parameters of these kernels are found using likelihood optimization

based on the camera-captured pixels. Consequently, the kernels harvest correlation over all dimensions and steer along the dimensions with highest correlation. As such they align with e.g. edges (in spatial dimensions) and temporal flow (in the time dimension) in the case of video. It is shown in this work that the SMoE representation is able to model images, 360-degree images, light fields, and light field video by using GMMs. One disadvantage of the limited parametrization of GMMs is that the linear nature of the resulting regressors might fail to capture high spatial frequencies such as noise and fine texture. Nevertheless, future models with more expressive regressors are not excluded.

Two challenges needed to be tackled in order to model immersive higher-dimensional image modalities. First, special care needs to be taken for spherical dimensions. Spherical dimensions are common in immersive image modalities, e.g. 360-degree video or circular light fields. The key insight here is that such coordinates can be reformulated to a Euclidean coordinate space. For example, 360° content with two spherical dimensions can be expressed as samples that lay on a unit sphere in a 3-D space. Furthermore, in this dissertation, it is shown that in such a case, the parameters of the model can still be regarded as 2-D using local per-kernel geometrical projections of the kernel parameters. As such, the translation to a 3-D space does not result in an increase of parameters per kernel. The second large challenge during the development of this framework is computational complexity when modeling immersive image modalities for two reasons. First, such image modalities yield billions of samples as the number of samples increases exponentially with increasing dimensionality. Secondly, the standard implementations of the employed algorithms scale badly with increasing number of samples and an increasing number of kernels. Therefore, important contributions of this work are the efficient modeling strategies that make use of the dense sample structure of the input data. The main observations are that kernels represent a relatively local patch of pixels and therefore localizing the operations during the modeling process can mitigate the otherwise infeasible computational demands of these algorithms.

In the last part of the dissertation, a coding method is proposed and evaluated on a range of image modalities in terms of rate-distortion (RD). As mentioned above, the proposed representation is compact as kernels can represent over tens of thousands of original image samples. The model thus consists solely of kernel parameters. Interestingly, when the dimensionality of the image modalities increases, the number of parameters necessary only grows quadratically per kernel, whereas the number of samples in dense representations grow exponentially. A coding scheme is thus presented in which the kernel parameters are binarized into a bitstream. The model parameters are first decorrelated both locally and globally. First, neighboring kernels have similar center values and thus allow for local decorrelation. Secondly, the spread of the kernels in the coordinate dimension tends to be repetitive over the whole model. The redundancy is captured by using a dictionary system that limits the number of possible kernel shapes. The decorrelated parameters are further encoded using an arithmetic coder.

The relative coding efficiency of SMoE compared to state-of-the-art techniques

increases when the dimensionality of the image modality increases. In images, kernels typically span between 8 and 100 pixels, whereas in light field video, a kernel can easily span 50,000 original pixels. For images, the SMoE method typically outperforms JPEG for low bitrates, but JPEG-2000 consistently outperforms the proposed method. For static 4-D light fields, the SMoE-based codec was outperformed by HEVC when using motion-compensation (low random access, complex decoding structure) in terms of PSNR and SSIM. Nevertheless, the SMoE-based codec did strongly outperform HEVC All-Intra (which allows similar granular random access as SMoE). Subjective tests were performed in order to assess view quality, view consistency, and refocusing after coding. These results show that SMoE is competitive with the best HEVC configuration up to the range of a MOS score above 4 (Perceptible but not annoying), arguably the most interesting range for coding schemes from a practical point of view. For 5-D light field video, it was found that the proposed approach can heavily outperform multiview-HEVC up to bitrate savings up to a factor of $4\times$.

The SMoE representation in this work is limited to linear regressors and thus assumes natural images to be able to be approximated as a smoothed piecewise linear function. However, the reality is that image modalities resemble more piecewise stationary functions and can exhibit high spatial frequencies in textured regions. With the current model, an infeasible number of small kernels would be necessary to capture all detail. Current work aimed and succeeded in proving feasibility to design a sparse information-rich representation that scales to any dimensionality with desired functionality for VR consumption, e.g. random access, inherent view interpolation, and pixel-parallel reconstructions. Future work will thus consist of introducing provisions for residual texture. However, in this work it is shown that even without these provisions, the proposed model is competitive for low-to-mid range bitrates while providing the functionality required for future CC-VR. Furthermore, the model is not trained to maximize PSNR of the reconstruction, but to maximize the likelihood of the model. As such, PSNR optimization could be a way to increase the RD-performance as early evidence shows. Finally, other properties and applications of SMoE need to be investigated and assessed.

It is clear that there is still a long way to go before we can enjoy the first full wide-range CC-VR applications. Nevertheless, this work provides a scalable framework that is future-proof to facilitate the road towards full CC-VR while showing alternative and competitive methods for representing and coding more common image modalities as well. In general, this work is not meant in any way to be the final solution, but to open a novel way of thinking in the development of image representations and to abandon some potentially outdated paradigms.

Nederlandse samenvatting –Summary in Dutch–

Met de introductie van analoge fotografie in de 19e eeuw konden mensen hun omgeving voor het eerst op film vastleggen. Later leidde fotografie tot analoge cinema door deze foto's als een snel bewegende reeks vast te leggen en opnieuw weer te geven. Tegenwoordig behoren afbeeldingen en video voornamelijk tot de digitale wereld en zijn beelden nog nooit zo alomtegenwoordig geweest. De digitalisering van beeldvorming maakte ook de digitale manipulatie van de beelden mogelijk, variërend van eenvoudige fotobewerking tot het inwerken van enorme hoeveelheden computergegenereerde beelden of "computer-generated imagery" (CGI) als special effects. Het domein van de computergrafiek heeft de afgelopen decennia enorme vooruitgang geboekt op de ontwikkeling van hulpmiddelen voor het maken van 3-D modellen, rendertechnologie en hardwareversnelling. Deze hulpmiddelen zijn niet alleen van toepassing voor het toevoegen van speciale effecten aan films, maar deze tools stellen ons ook in staat om complete virtuele werelden te creëren. Gamers kunnen bijvoorbeeld door grote gebieden dwalen en objecten met een adembenemende kwaliteit ver weg of dichtbij bekijken. Nog beter, ze kunnen zelfs fysisch omgaan met de objecten! Deze concepten waren zelfs vijftig jaar geleden nog volledig ondenkbaar.

Merk op dat de filmervaring van de kijker eigenlijk niet fundamenteel veranderd is door de digitalisering. Namelijk, een filmregisseur legt één specifiek standpunt vast, daarna wordt dit standpunt terug getoond aan de kijker. De regisseur heeft dus volledige controle over de ervaring van de kijker. Nu, stel je voor dat je zelf het verhaal kunt binnenstappen terwijl je de scène met fotorealistische kwaliteit kunt aanschouwen. Hierdoor kan je jezelf verplaatsen in de ruimte en zelf de ruimte achter objecten verkennen. Grote evenementen kunnen vervolgens op een zodanige manier worden vastgelegd dat de kijker het evenement vanuit elk standpunt als een stille vrijrondlopende toeschouwer kan herbeleven. In een dergelijk geval kunnen kijkers genieten van de navigatievrijheid zoals bij het gamen, maar dan met het beeldechtheid van cinema. Toepassingen zijn overigens niet beperkt tot entertainment of het maken van "levende herinneringen". Het is gemakkelijk om toepassingen voor te stellen in bijvoorbeeld het onderwijs, cultureel erfgoed, therapie, afstandsbediening in de industrie, geneeskunde, toerisme, enz.

Niettemin is virtual reality (VR) voor camera-opgenomen scènes fundamenteel anders en complexer dan in vergelijking met de VR-beleving van computergegenereerde (CG) scènes (zoals bijvoorbeeld bij gaming). Het probleem is veel

uitdagender vanwege het gebrek aan geometrische kennis van de scène. Momenteel is 360-graden video de voornaamste vorm van camera-opgenomen virtual reality, of "camera-captured virtual reality" (CC-VR), maar die is onvoldoende voor het leveren van een volledige VR-ervaring omdat het geen positionele vrijheid aan de gebruiker levert. Kijkers kunnen alleen rondkijken, maar de locatie van hun hoofd staat vast. Als zodanig is er een grote interesse om deze beperkingen te overwinnen en aldus de kijker de mogelijkheid te geven om in een grote complexe ruimte tussen objecten te kunnen navigeren en zelfs door deuren te kunnen lopen.

In dit proefschrift presenteer ik een nieuwe methode voor het weergeven en coderen van een breed scala aan beeldmodaliteiten (bijv. afbeeldingen, video, lichtvelden) die op een toekomstbestendige manier is ontworpen om toekomstige CC-VR mogelijk te maken. Eerst worden traditionele methoden besproken die momenteel gepivoteerd worden naar CC-VR en vervolgens worden functionele eisen geïdentificeerd voor een VR-klare beeldrepresentatie. Ten tweede wordt de nieuwe representatie geïntroduceerd en geïllustreerd voor lager-dimensionale afbeeldingen (bij. afbeeldingen) en vervolgens verder toegepast op meer immersieve beeldmodaliteiten (bijv. 360-graden afbeeldingen en lichtvelden). Ten slotte wordt een coderingsmethode gepresenteerd en geëvalueerd om camera-vastgelegde scènes efficiënt op te slaan, te verzenden en uit te zenden.

Op dit moment worden twee algemene strategieën gevolgd om immersieve ervaringen van camera-vastgelegde scènes mogelijk te maken. De eerste strategie bestaat uit het construeren van een 3D-model van de scène. De stappen voor geometrie en textuurbenadering zijn echter inherent verlieshebbend, tijdrovend en vereisen vaak handmatige interventie wanneer de kwaliteit fotorealistisch moet zijn. Bovendien worstelen ze met niet-rigide objecten zoals rook, vuur, water en transparante oppervlakken. De tweede strategie is gebaseerd op bekende videocoderingsmethoden. Volgens deze filosofie worden scènes weergegeven door een aantal camera-standpunten te coderen als video, en de ontbrekende standpunten te reconstrueren door beeldsynthese. Videocoderingsmethoden bieden uitstekende codeerefficiëntie voor video en voor problemen die kunnen worden vertaald naar video. Op de lange termijn kan echter worden gesteld dat deze probleemvertaling minder evident wordt voor VR met uitgebreide gebruikersautonomie. Bij videocodering worden video's weergegeven als een reeks stilstaande beelden. Deze afbeeldingen (of "frames") hebben dus een expliciete volgorde, nl. tijd. Deze logische volgorde was de grootste bron voor het verminderen van opslagruimte voor video's. Twee opeenvolgende frames lijken veel op elkaar. Als één frame eenmaal bekend is, hoeft alleen het verschil tussen dat frame en het volgende frame te worden opgeslagen of verzonden. Deze paradigma's worden echter uitgedaagd als de kijker op elk moment mag kiezen waar hij wil kijken. Met andere woorden, de volgorde van de frames is niet bekend tijdens de coderingstijd, maar wordt alleen bepaald bij het afspelen door de beweging van de gebruiker. Bovendien groeit het aantal mogelijke weergaven exponentieel met het niveau van de autonomie van de gebruiker. Daarbovenop heeft het vereiste beeldsyntheseproces om de ontbrekende beelden te genereren beperkingen (bijv. bij het omgaan met reflecties en occlusies).

De voorgestelde *Steered Mixture-of-Experts* (SMoE)-methodologie modelleert de lichtinformatie als hoger-dimensionale data. De reden is dat de waargenomen 2-D beelden in een VR-scène op elke positie en blikoriëntatie 2-D segmenten van hoger-dimensionale lichtgegevens zijn. Het idee van de voorgestelde strategie is om hoger-dimensionale data als zodanig te behandelen en de gegevens niet te beperken tot 2-D bemonsteringsroosters of om de geometrie van een scène te construeren. Meer specifiek wordt de onderliggende pixel-genererende plenoptische functie benaderd door inherent hoger-dimensionale atomen te gebruiken, de zogenaamde kernen of "kernels". Een kern kan worden gezien als een multidimensionale generalisatie van de pixel. Deze kernels zorgen voor het gelijktijdig oogsten van pixelkleurcorrelatie in verschillende richtingen: b.v. tijd, pixelpositie, camerapositie. Verder beschrijft een enkele kern een multidimensionale gradiënt langs deze dimensies.

De voordelen van de voorgestelde SMoE-representatie zijn als volgt. Ten eerste is de methode schaalbaar in dimensionaliteit en kan deze dus worden toegepast op beeldmodaliteiten met willekeurige dimensionaliteit (bijv. van beelden tot lichtvelden). Ten tweede kan een enkele kernel een groot aantal oorspronkelijk vastgelegde pixels dekken, terwijl het aantal parameters per kernel beperkt blijft. Daardoor is de representatie geschikt voor het toepassen van compressie. Ten derde is het SMoE-model continu, dus het reconstrueren van een beeld komt erop neer dat het model alleen moet bemonsterd worden met een 2-D raster met de gewenste resolutie. Ten vierde kan de reconstructie van 2-D beelden op basis van het model worden uitgevoerd op een lichte en pixel-niveau parallelle manier. Ten vijfde, zijn kernels niet van elkaar afhankelijk. Om slechts een deel van de beeldmodaliteit te reconstrueren hoeven enkel de kernels in die bepaalde omgeving van de coördinatenruimte aanwezig te zijn. Dit zorgt voor een fijnmazige "random access" bij het streamen van natuurlijke VR-scènes. Ten slotte neemt de representatie de structuur van de gegevens zelf over. Als zodanig onthullen de parameters van het model relevante machineleesbare informatie over de scène (bijv. randen, intensiteitsstroom, bewegingsstroom, ...) en kan zelfs impliciet de geometrie van een scène aangegeven worden in het geval van lichtvelden. Bovendien biedt het model inherent een multidimensionale segmentatie van de beeldmodaliteit.

De SMoE-methode is gebaseerd op de gegevensadaptieve verdeling van de coördinaatruimte van de beeldmodaliteit. Voor elke afbeeldingsmodaliteit wordt de coördinaatruimte gedefinieerd als $\mathbb{R}^p$ en de kleurruimte als $\mathbb{R}^q$. Voor afbeeldingen, video, lichtveldafbeeldingen en lichtveldvideo is de dimensionaliteit p respectievelijk 2, 3, 4 en 5. Voor monochrome afbeeldingen is q 1 en voor kleurenafbeeldingen is q doorgaans 3 . Het doel is om de coördinaatruimte $\mathbb{R}^p$ te verdelen in stationaire regio's en regressors te vinden ($f : \mathbb{R}^p \mapsto \mathbb{R}^q$) die lokaal deze stationaire regio goed benaderen.

De onderliggende veronderstelling is dat de pixels instantiaties zijn van een niet-linear of niet-stationair willeurig proces dat kan worden gemodelleerd door ruimtelijke, stuksgewijs stationaire, stochastische processen. Als zodanig houdt het model rekening met verschillende delen van de afbeelden en hun segmentatieranden. Bovendien bestaan de beelden in het epipolaire vlak in lichtvelden uit

diagonaal gestructureerde lijnen, en in video kan beweging worden benaderd door lijnsegmenten zoals dat wordt gedaan in bewegingscompensatie bij traditionele videocodering. Daarom wordt ook in dit proefschrift een stuksgewijze representatie van beeldmodaliteiten nagestreefd. Elk dergelijk stationair gebied wordt dan idealiter weergegeven door één enkele kernel. Een dergelijke verdeling van de inputruimte is een strategie in machinaal leren dat beter gekend is als een *Mixture-of-Experts* methode. SMoE is echter gebaseerd op een specifieke versie van de MoE-benadering, meer bepaald op basis van mengselmodellen or "mixture models". In deze versie zijn zowel de segmentatie alsook de lokale regressors (de *experten*) afgeleid van de $(p+q)$-dimensionale componenten van een mengselmodel. Dit mengselmodel modelleert de gezamelijke waarschijnlijkheidsverdeling van een coördinaatvector $X \in \mathbb{R}^p$ en een kleurvector $Y \in \mathbb{R}^q$. Voor grijswaardenafbeeldingen is het model dus 3-D (twee ruimtelijke coördinaten en één kleurwaarde), terwijl voor kleurenlichtvelden het model 7-D is (vier coördinaten en drie kleurwaarden).

In dit proefschrift ligt de nadruk op de geval waar het mengselmodel bestaat uit Gaussiaanse distributies, een *Gaussian Mixture Model* (GMM). Eén Gausskernel in het model definieert een lineaire regressor door de verwachtingswaarde van de geconditioneerde $Y|X$ distributie. Verder definiëren alle kernels samen een segmentatie van de coördinatenruimte. Het model bestaat dus enkel uit een reeks Gaussiaanse kernels die worden gedefiniëerd door hun centra en covarianties. De reden om voor GMM's te kiezen is dat ze elegante wiskunde en beperkte parametrisering bieden. De MoE op basis van GMM's resulteert in stuksgewijze lineaire benaderingen. Om een beeld te modelleren, worden de parameters van deze kernels gevonden met behulp van waarschijnlijkheidsoptimalisatie op basis van de door camera-opgenomen pixels. Bijgevolg oogsten de kernels correlatie over alle dimensies en strekken ze zichzelf uit langs de dimensies met de hoogste correlatie. Als zodanig komen ze overeen met bijv. randen (in ruimtelijke dimensies) en temporele stroming (in de tijdsdimensie) in het geval van video. In dit werk wordt aangetoond dat de SMoE-weergave afbeeldingen, 360-graden afbeeldingen, lichtvelden en lichtveldvideo kan modelleren met behulp van GMM's. Een nadeel van de beperkte parametrisering van GMM's is dat de lineaire aard van de resulterende regressors er mogelijk niet in slaagt om hoge ruimtelijke frequenties zoals ruis en fijne textuur vast te leggen. Niettemin worden toekomstige modellen met meer expressieve regressoren niet uitgesloten.

Twee uitdagingen moesten worden aangepakt om immersieve, hogerdimensionale beeldmodaliteiten te modelleren. Ten eerste moet speciale aandacht worden besteed aan sferische dimensies. Sferische dimensies zijn vaak voorkomend in immersieve beeldmodaliteiten, b.v. 360-graden video of cirkelvormige lichtvelden. Het belangrijkste inzicht hier is dat dergelijke coördinaten kunnen worden geherformuleerd tot een Euclidische coördinaatruimte. 360-graden beelden met twee sferische dimensies kan bijvoorbeeld worden uitgedrukt als monsters die op een eenheidsbol in een 3-D ruimte liggen. Verder is in dit proefschrift aangetoond dat in een dergelijk geval de parameters van het model nog steeds kunnen worden beschouwd als 2-D met behulp van lokale per-kernel geometrische

projecties van de kernelparameters. Als zodanig leidt de vertaling naar een 3-D ruimte niet tot een toename van parameters per kernel. De tweede grote uitdaging tijdens de ontwikkeling van dit raamwerk is computationele complexiteit bij het modelleren van immersieve beeldmodaliteiten om twee redenen. Ten eerste leveren dergelijke beeldmodaliteiten miljarden monsters op, aangezien het aantal monsters exponentieel toeneemt met toenemende dimensionaliteit. Ten tweede, de standaardimplementaties van de gebruikte algoritmen schalen slecht met een toenemend aantal monsters en een toenemend aantal kernels. Daarom zijn belangrijke bijdragen van dit werk de efficiënte modelleringsstrategieën die gebruik maken van de dichte pixelstructuur van de invoerbeelden. De belangrijkste waarnemingen zijn dat kernels een relatief lokale regio vertegenwoordigen en dat daarom het lokaliseren van de bewerkingen tijdens het modelleringsproces de anders onhaalbare rekenbehoeften van deze algoritmen kan verminderen.

In het laatste deel van het proefschrift wordt een coderingsmethode voorgesteld en geëvalueerd op een reeks beeldmodaliteiten uitgedrukt als bestandsgrootte versus beelddistortie of "rate-distortion" (RD). Zoals hierboven vermeld, is de voorgestelde weergave compact omdat de kernen meer dan tienduizenden originele pixels kunnen vertegenwoordigen. Het model bestaat dus uitsluitend uit kernelparameters. Interessant is dat wanneer de dimensionaliteit van de beeldmodaliteiten toeneemt, het aantal benodigde parameters alleen kwadratisch groeit per kernel, terwijl het aantal monsters in dichte representaties exponentieel groeit. Daarom wordt ook een coderingsschema gepresenteerd in dit werk waarin de kernelparameters in een bitstroom worden gebinariseerd. De modelparameters worden eerst zowel lokaal als globaal gedecorreleerd. Ten eerste hebben naburige kernels vergelijkbare centrumwaarden en laten dus lokale decorrelatie toe. Ten tweede is de verspreiding van de kernels in de coördinaatdimensie vaak repetitief over het hele model. De redundantie wordt opgevangen met behulp van een woordenboeksysteem dat het aantal mogelijke kernelvormen beperkt. De gedecorreleerde parameters worden verder gecodeerd met behulp van een rekenkundige codeerder.

De relatieve codeerefficiëntie van SMoE in vergelijking met de state-of-the-art technieken neemt toe naarmate de dimensionaliteit van de beeldmodaliteit toeneemt. In afbeeldingen beslaan kernels meestal 8 tot 100 pixels, terwijl in lichtveldvideo een kernel gemakkelijk 50.000 originele pixels kan omvatten. Voor 2-D afbeeldingen presteert de SMoE-methode doorgaans beter dan JPEG voor lage bitsnelheden, maar JPEG-2000 presteert consistent beter dan de voorgestelde methode. Voor statische 4-D lichtvelden werd de op SMoE gebaseerde codec overtroffen door HEVC bij gebruik van bewegingscompensatie (lage willekeurige toegang, complexe decodeerstructuur) uitgedrukt in PSNR en SSIM. Desalniettemin presteerde de op SMoE gebaseerde codec sterk beter dan HEVC All-Intra (waardoor vergelijkbare fijnmazige random access mogelijk is zoals in SMoE). Subjectieve tests werden uitgevoerd om de beeldkwaliteit, de beeldconsistentie en het refocussen na codering te beoordelen. Deze resultaten laten zien dat SMoE concurreert met de beste HEVC-configuratie tot het bereik van een MOS-score boven de 4 (waarneembaar maar niet vervelend), misschien wel het meest interessante bereik voor coderingsschema's vanuit praktisch oogpunt. Voor 5-D lichtveldvi-

deo werd besloten dat de voorgestelde aanpak multiview-HEVC aanzienlijk kan overtreffen, zelf met een bitrate reductiefactor tot $4\times$.

De SMoE-representatie in dit werk is beperkt tot lineaire regressoren en veronderstelt dus dat natuurlijke beelden kunnen worden benaderd als een afgevlakte, in stukjes verdeelde lineaire functie. De realiteit is echter dat beeldmodaliteiten meer op stuksgewijze stationaire functies lijken en daarbij hoge ruimtelijke frequenties in gestructureerde gebieden kunnen vertonen. Met het huidige model zou een onhaalbaar groot aantal kleine kernels nodig zijn om alle details vast te leggen. Het huidige werk was gericht op en slaagde erin de haalbaarheid te bewijzen van het ontwerpen van een compacte en informatierijke representatie die schaalt naar elke dimensionaliteit met gewenste functionaliteit voor VR-consumptie, b.v. random access, inherente beeldinterpolatie en pixel-parallelle reconstructies. Toekomstig werk zal dus bestaan uit het invoeren van voorzieningen voor residuele textuur. In dit werk is echter aangetoond dat zelfs zonder deze voorzieningen het voorgestelde model concurrerend is voor lage tot middelgrote bitrates en tegelijkertijd de functionaliteit biedt die vereist is voor toekomstige CC-VR. Bovendien is het model niet geoptimaliseerd om de PSNR van de reconstructie te maximaliseren, maar om de waarschijnlijkheid van het model te maximaliseren. Als zodanig kan PSNR-optimalisatie een manier zijn om de RD-prestaties te verbeteren, zoals uit vroeg bewijs blijkt. Ten slotte moeten andere eigenschappen en toepassingen van SMoE worden onderzocht en beoordeeld.

Het is duidelijk dat er nog een lange weg te gaan is voordat we kunnen genieten van de eerste uitgestrekte CC-VR-toepassingen. Desalniettemin biedt dit werk een schaalbaar en toekomstbestendig raamwerk dat de weg naar volledige CC-VR toepassingen kan vergemakkelijken. Het is op geen enkele manier bedoeld dat dit proefschrift de ultieme oplossing geeft, maar dit werk moedigt een nieuwe manier van denken aan die moet gevolgd worden bij de ontwikkeling van nieuwe beeldrepresentaties en waarbij ook mogelijks achterhaalde paradigma's achterwege gelaten moeten worden.

Deutsche Zusammenfassung –Summary in German–

Die Einführung der analogen Fotografie im 19. Jahrhundert ermöglichte es den Menschen, ihre Umgebung zum ersten Mal auf Film festzuhalten. Später führte die Fotografie zum analogen Kino, indem sie Fotografien in einer sich schnell bewegenden Sequenz aufzeichnete und anzeigte. Heutzutage befinden sich Bilder und Videos hauptsächlich in der digitalen Welt und waren noch nie so allgegenwärtig. Die Digitalisierung der Bildgebung ermöglichte ferner die digitale Bearbeitung der Bilder, von der Fotobearbeitung bis hin zur Einbeziehung großer Mengen computergenerierter Bilder (engl. *computer generated imagery*, CGI). In den letzten Jahrzehnten hat die Computergrafik enorme Fortschritte bei der Erstellung von Inhalten, der Rendertechnologie und der Hardwarebeschleunigung erzielt. Diese Werkzeuge können nicht nur Filme mit Spezialeffekten versehen, sondern ermöglichen auch die Erstellung ganzer digitaler Welten. Zum Beispiel können Spieler über große Gebiete streifen und Objekte in der Ferne oder in der Nähe mit atemberaubender Qualität betrachten. Noch besser, sie können mit den Objekten interagieren! Diese Konzepte wären noch vor fünfzig Jahren unglaublich schwer vorstellbar gewesen.

Interessanterweise hat die Digitalisierung von Bildgebung und Kino nichts an der Art und Weise geändert, in der der Betrachter von der Kamera erfasste visuelle Daten wahrnimmt. Ein Filmregisseur nimmt einen bestimmten Blickwinkel auf, woraufhin den Zuschauern gezeigt wird, was der Regisseur ihnen zeigen wollte. Stellen Sie sich nun vor, Sie betreten diese Geschichte mit fotorealistischer Qualität. Auf diese Weise können Sie sich bewegen und den Raum hinter Objekten erkunden. Großereignisse könnten dann so erfasst werden, dass der Zuschauer das Ereignis aus jedem Blickwinkel als stiller Zuschauer im Freien nacherleben kann. In einem solchen Fall können die Zuschauer die Navigationsfreiheit wie beim Spielen genießen, jedoch mit dem Fotorealismus des Kinos. Anwendungen sind nicht auf Unterhaltung oder das Erstellen von “lebenden Erinnerungen” beschränkt. Es ist leicht vorstellbar, Anwendungen in den Bereichen Bildung, Erhaltung des kulturellen Erbes, Therapie, Fernsteuerung in Industrie, Medizin, Tourismus usw. zu finden.

Nichtsdestotrotz unterscheidet sich die virtuelle Realität (VR) für mit einer Kamera aufgenommene Szenen grundlegend von der VR-Nutzung computergenerierter (CG) Szenen (z. B. bei Computerspiele) und ist komplexer. Das Problem ist aufgrund des Mangels an geometrischen Kenntnissen viel schwieriger.

Gegenwärtig ist die von einer Kamera erfasste virtuelle Realität (engl. *camera-captured virtual reality* , CC-VR) hauptsächlich auf 360-Grad-Video beschränkt, was für die Bereitstellung einer vollständigen VR-Erfahrung unzureichend ist, da es dem Benutzer keine Positionsfreiheit bietet. Der Betrachter kann sich nur umschauen, aber die Position seines Kopfes ist festgelegt. Daher besteht ein großes Interesse daran, diese Einschränkungen zu überwinden und es dem Betrachter zu ermöglichen, in einer Szene über eine große räumliche Ausdehnung zwischen Objekten und sogar durch Türen zu gehen.

In dieser Dissertation stelle ich eine neuartige Methode zur Darstellung und Codierung einer Vielzahl von Bildmodalitäten (z. B. Bilder, Videos, Lichtfelder) vor, die zukunftssicher gestaltet ist, um zukünftige CC-VR zu ermöglichen. Zunächst werden traditionelle Methoden diskutiert, die derzeit auf CC-VR ausgerichtet sind, und Anforderungen an eine VR-fähige Bilddarstellung ermittelt. Zweitens wird die neue Darstellung für Bilder mit niedrigeren Dimensionen (z. B. Bilder) eingeführt und dargestellt und dann auf immersivere Bildmodalitäten (z. B. 360-Grad-Bilder und Lichtfelder) angewendet. Schließlich wird eine Codierungsmethode vorgestellt und evaluiert, um mit der Kamera aufgenommene Szenen effizient zu speichern, zu übertragen und zu senden.

Gegenwärtig werden zwei allgemeine Strategien verfolgt, um ein echtes, immersives Erlebnis von mit der Kamera aufgenommenen Szenen zu ermöglichen. Die erste Strategie besteht darin, ein 3D-Modell der Szene zu erstellen. Die Schritte der Geometrie- und Texturapproximation sind jedoch von Natur aus verlustreich, zeitaufwendig und erfordern oft manuelle Eingriffe, wenn die Qualität fotorealistisch sein soll. Darüber hinaus kämpfen sie mit nicht starren Objekten wie Rauch, Feuer, Wasser und transparenten Oberflächen. Die zweite Strategie beruht auf bekannten Videocodierungsverfahren. Nach dieser Philosophie werden Szenen dargestellt, indem eine Reihe von Kamera-Blickwinkeln als Video codiert und die fehlenden durch Blicksynthese weiter rekonstruiert werden. Videocodierungsmethoden bieten eine hervorragende Codierungsleistung für Videos und für Probleme, die in Videos übersetzt werden können. Langfristig kann jedoch argumentiert werden, dass diese Problemübersetzung für VR mit umfassender Benutzerautonomie weniger offensichtlich wird. Bei der Videokodierung werden Videos als Folge von Standbildern dargestellt. Diese Bilder (oder “Frames”) haben somit eine explizite zeitliche Reihenfolge. Diese logische Reihenfolge war der größte Ursprung für die Reduzierung des Speicherplatzes für Videos. Zwei aufeinanderfolgende Frames sind sich normalerweise sehr ähnlich. Sobald ein Frame bekannt ist, muss nur die Differenz zwischen diesem Frame und dem nächsten Frame gespeichert oder übertragen werden. Diese Paradigmen werden jedoch in Frage gestellt, sobald der Betrachter jederzeit auswählen kann, wohin er schauen möchte. Mit anderen Worten, die Reihenfolge der Frames ist zum Zeitpunkt der Codierung nicht bekannt, sondern wird erst bei der Wiedergabe durch die Bewegung des Benutzers festgelegt. Darüber hinaus wächst die Anzahl der möglichen Ansichten exponentiell mit der Autonomie des Benutzers. Darüber hinaus weist der erforderliche Ansichtssynthesevorgang zum Erzeugen der fehlenden Ansichten Einschränkungen auf (z. B. beim Umgang mit Reflexionen und Verdeckungen).

Die hier gezeigte SMoE-Methode (*Steered Mixture-of-Experts*) modelliert die Lichtinformationen als höherdimensionale Daten. Der Grund dafür ist, dass die beobachteten 2-D-Ansichten in einer VR-Szene an jeder Position und Blickrichtung 2-D-Schichten höherdimensionaler Lichtdaten sind. Die Idee der vorgestellten Strategie besteht darin, höherdimensionale Daten als solche zu behandeln und die Daten nicht auf 2-D-Abtastgitter zu reduzieren oder die 3-D-Geometrie einer Szene zu konstruieren. Insbesondere wird die zugrunde liegende pixelerzeugende plenoptische Funktion unter Verwendung von inhärent höherdimensionalen Atomen, die als "Kernel"bezeichnet werden, angenähert. Ein Kernel kann als mehrdimensionale Verallgemeinerung des Pixels angesehen werden. Diese Kerne ermöglichen das gleichzeitige Erfassen der Pixelfarbkorrelation in verschiedene Richtungen: z.B. Zeit, Pixelposition, Kameraposition. Darüber hinaus beschreibt ein einzelner Kernel einen mehrdimensionalen Gradienten entlang dieser Dimensionen.

Die Vorteile der vorgestellten SMoE-Darstellung sind wie folgt. Erstens ist das Verfahren in seiner Dimensionalität skalierbar und kann daher auf beliebig dimensionale Bildmodalitäten angewendet werden. Zweitens kann ein einzelner Kernel eine große Anzahl ursprünglich erfasster Pixel abdecken, während die Anzahl der Parameter pro Kernel begrenzt bleibt. Daher ist die Darstellung für die Anwendung von Kompression geeignet. Drittens ist das SMoE-Modell kontinuierlich, daher läuft die Rekonstruktion einer Ansicht darauf hinaus, das Modell lediglich mit der gewünschten Auflösung abzutasten. Viertens kann die Rekonstruktion von 2-D-Ansichten basierend auf dem Modell auf eine leichte und pixelgenaue Weise parallel ausgeführt werden. Fünftens sind die Kerne nicht voneinander abhängig, um einen Teil der Bildmodalität zu rekonstruieren, müssen nur die Kerne in der Nähe dieses Teils des Koordinatenraums vorhanden sein. Dies ermöglicht einen fein abgestimmten wahlfreien Zugriff beim Streamen von natürlichen VR-Szenen. Schließlich übernimmt die Darstellung die Struktur der Daten selbst. Somit enthüllen die Parameter des Modells relevante maschinenlesbare Informationen über die Szene (z. B. Kanten, Intensitätsfluss, Bewegungsfluss, ...) und können im Fall von Lichtfeldern sogar implizit die Geometrie einer Szene angeben. Darüber hinaus liefert das Modell von Natur aus eine mehrdimensionale Segmentierung der Bildmodalität.

Die SMoE-Methode basiert auf der datenadaptiven Aufteilung des Koordinatenraums der Bildmodalität. Für jede Bildmodalität ist der Koordinatenraum als $\mathbb{R}^p$ und der Farbraum als $\mathbb{R}^q$ definiert. Für Bilder, Videos, Lichtfeldbilder und Lichtfeldvideos beträgt die Dimensionalität p jeweils 2, 3, 4 und 5. Für monochrome Bilder beträgt q 1 und für Farbbilder beträgt q typischerweise 3 . Das Ziel ist es, den Koordinatenraum $\mathbb{R}^p$ in stationäre Regionen aufzuteilen und Regressoren ($f : \mathbb{R}^p \mapsto \mathbb{R}^q$) zu finden, die diese stationären Regionen lokal gut approximieren. Die zugrunde liegende Annahme ist, dass Bildpixel Instanzen eines nichtlinearen oder nichtstationären Zufallsprozesses sind, die durch räumlich-stückweise stationäre stochastische Prozesse modelliert werden können. Als solches berücksichtigt das Modell verschiedene Bereiche des Bildes und deren Segmentierungsgrenzen. Darüber hinaus bestehen die Epipolarebenenbilder in Licht-

feldern aus diagonal strukturierten Linien, und in Video kann die Bewegung durch Liniensegmente angenähert werden, wie dies bei der Bewegungskompensation bei der herkömmlichen Videokodierung der Fall ist. Daher wird in dieser Dissertation eine stückweise Annäherung der Bildmodalitäten verfolgt. Jede solche stationäre Region wird dann idealerweise durch einen einzelnen Kernel dargestellt. Eine solche Aufteilung des Eingaberaums ist eine allgemeine *Mixture-of-Experts* (MoE) -Strategie, die auf dem Gebiet des maschinellen Lernens bekannt ist. SMoE basiert jedoch auf einer Mischmodellversion (oder einer alternativen Version) des MoE-Ansatzes. In dieser Version werden sowohl Segmentierungs- als auch lokale Regressoren (die *Experten*) aus den $(p + q)$-dimensionalen Komponenten eines Mischungsmodells abgeleitet: Dieses Mischungsmodell modelliert die gemeinsame Wahrscheinlichkeitsverteilung eines Probenkoordinatenvektors $X \in \mathbb{R}^p$ und eines Probefarbvektors $Y \in \mathbb{R}^q$. Für Graustufenbilder ist das Modell somit 3-D (zwei räumliche Koordinaten, ein Farbwert), während für Farblichtfelder das Modell 7-D ist (vier Koordinaten, drei Farbwerte).

In dieser Dissertation liegt der Schwerpunkt auf dem Fall von einem *Gaussian Mixture Model* (GMM). Ein Gaußscher Kernel im Modell definiert einen linearen Regressor durch die Bedingung $Y|X$, und alle Kernel zusammen definieren eine Segmentierung des Koordinatenraums. Das Modell besteht also nur aus einer Menge von Gaußschen Kernen, die durch ihre Zentren und Kovarianzen definiert sind. Der Grund für die Wahl von GMMs ist, dass sie elegante Mathematik und begrenzte Parametrisierung bieten. Das auf GMMs basierende MoE führt zu geglätteten stückweisen Approximationen. Um eine Bildmodalität zu modellieren, werden die Parameter dieser Kernel unter Verwendung einer Wahrscheinlichkeitsoptimierung basierend auf den von der Kamera erfassten Pixeln gefunden. Folglich sammeln die Kerne Korrelationen über alle Dimensionen und steuern entlang der Dimensionen mit der höchsten Korrelation. Als solche richten sie sich z.B. an Kanten (in räumlichen Dimensionen) und im zeitlichen Fluss (in zeitlichen Dimensionen) im Fall von Video. In dieser Arbeit wird gezeigt, dass die SMoE-Darstellung Bilder, 360-Grad-Bilder, Lichtfelder und Lichtfeldvideos mithilfe von GMMs modellieren kann. Ein Nachteil der eingeschränkten Parametrisierung von GMMs ist, dass die lineare Natur der resultierenden Regressoren möglicherweise keine hohen räumlichen Frequenzen wie Rauschen und feine Textur erfasst. Trotzdem sind zukünftige Modelle mit ausdrucksstärkeren Regressoren nicht ausgeschlossen.

Zwei Herausforderungen mussten angegangen werden, um immersive, höherdimensionale Bildmodalitäten zu modellieren. Erstens müssen die sphärischen Dimensionen besonders betrachtet werden. Sphärische Dimensionen sind bei immersiven Bildmodalitäten üblich, z.B. 360-Grad-Video oder kreisförmige Lichtfelder. Die wichtigste Erkenntnis hier ist, dass solche Koordinaten in einen euklidischen Koordinatenraum umformuliert werden können. Beispielsweise kann ein 360-Grad-Inhalt mit zwei sphärischen Dimensionen als Punkte ausgedrückt werden, die auf einer Einheitskugel in einem 3-D-Raum liegen. Darüber hinaus wird in dieser Dissertation gezeigt, dass in einem solchen Fall die Parameter des Modells unter Verwendung lokaler geometrischer Projektionen

der Kernparameter pro Kern als 2-D betrachtet werden können. Daher führt die Übersetzung in einen 3-D-Raum nicht zu einer Erhöhung der Parameter pro Kernel. Die zweite große Herausforderung bei der Entwicklung dieses Frameworks ist die Komplexität der Berechnungen bei der Modellierung immersiver Bildmodalitäten aus zwei Gründen. Erstens ergeben solche Bildmodalitäten Milliarden von Abtastwerten, wenn die Anzahl der Abtastwerte mit zunehmender Dimension exponentiell zunimmt. Zweitens sind die Standardimplementierungen der verwendeten Algorithmen mit zunehmender Anzahl von Abtastwerten und einer zunehmenden Anzahl von Kerneln schlecht skalierbar. Wichtige Beiträge dieser Arbeit sind daher die effizienten Modellierungsstrategien, die die dichte Stichprobenstruktur der Eingabedaten nutzen. Die wichtigsten Beobachtungen sind, dass Kernel ein relativ lokales Pixel-Patch darstellen und daher die Lokalisierung der Operationen während des Modellierungsprozesses die ansonsten nicht realisierbaren Rechenanforderungen dieser Algorithmen mindern kann.

Im letzten Teil der Dissertation wird eine Codierungsmethode vorgeschlagen und anhand einer Reihe von Bildmodalitäten hinsichtlich der Ratenverzerrung (RD) bewertet. Wie oben erwähnt, ist die vorgeschlagene Darstellung kompakt, da Kernel über Zehntausende von Originalbildproben darstellen können. Das Modell besteht also ausschließlich aus Kernelparametern. Interessanterweise wächst die Anzahl der erforderlichen Parameter nur quadratisch pro Kern, wenn die Dimensionalität der Bildmodalitäten zunimmt, während die Anzahl der Abtastwerte in dichten Darstellungen exponentiell zunimmt. Somit wird ein Codierungsschema vorgestellt, bei dem die Kernelparameter in einen Bitstrom binärisiert werden. Die Modellparameter werden zunächst lokal und global dekorreliert. Erstens haben benachbarte Kernel ähnliche Mittelwerte und ermöglichen so eine lokale Dekorrelation. Zweitens tendiert die Ausbreitung der Kerne in der Koordinatendimension dazu, sich über das gesamte Modell zu wiederholen. Die Redundanz wird mithilfe eines Wörterbuchsystems erfasst, das die Anzahl der möglichen Kernelformen begrenzt. Die dekorrelierten Parameter werden unter Verwendung eines arithmetischen Codierers weiter codiert.

Die relative Codiereffizienz von SMoE im Vergleich zu aktuellen Techniken nimmt zu, wenn die Dimensionalität der Bildmodalität zunimmt. In Bildern erstrecken sich die Kernel normalerweise über einen Bereich zwischen 8 und 100 Pixel, wohingegen ein Kernel in Lichtfeldvideos problemlos 50.000 Originalpixel überspannen kann. Bei Bildern übertrifft die SMoE-Methode JPEG bei niedrigen Bitraten in der Regel, JPEG-2000 übertrifft die vorgestellte Methode jedoch durchweg. Bei statischen 4-D-Lichtfeldern wurde der SMoE-basierte Codec von HEVC bei Verwendung der Bewegungskompensation (geringer wahlfreier Zugriff, komplexe Decodierungsstruktur) in Bezug auf PSNR und SSIM übertroffen. Trotzdem hat der SMoE-basierte Codec HEVC All-Intra (der einen ähnlich granularen Direktzugriff wie SMoE ermöglicht) deutlich übertroffen. Subjektive Tests wurden durchgeführt, um die Ansichtsqualität, die Ansichtskonsistenz und die Neuausrichtung nach dem Codieren zu bewerten. Diese Ergebnisse zeigen, dass SMoE mit der besten HEVC-Konfiguration bis zu einem MOS-Wert über 4 (wahrnehmbar, aber nicht störend) wettbewerbsfähig ist, was aus praktischer Sicht

wahrscheinlich der interessanteste Bereich für Codierungsschemata ist. Bei 5-D-Lichtfeldvideos wurde festgestellt, dass der vorgestellte Ansatz die Leistung von Multiview-HEVC deutlich übertreffen kann, um Bitrateneinsparungen von bis zu dem Faktor 4 zu erzielen.

Die SMoE-Darstellung in dieser Arbeit ist auf lineare Regressoren beschränkt und geht daher davon aus, dass natürliche Bilder als geglättete stückweise lineare Funktion angenähert werden können. Die Realität ist jedoch, dass Bildmodalitäten mehr stückweise stationären Funktionen ähneln und in texturierten Bereichen hohe räumliche Frequenzen aufweisen können. Mit dem aktuellen Modell wäre eine unüberschaubare Anzahl kleiner Kernel erforderlich, um alle Details zu erfassen. Gegenwärtige Arbeiten zielten darauf ab und es gelang ihnen, die Machbarkeit zu beweisen, eine Darstellung mit geringer Informationsfülle zu entwerfen, die mit der gewünschten Funktionalität für den VR-Verbrauch auf jede Dimension skalierbar ist, z.B. Direktzugriff, inhärente Ansichtsinterpolation und pixelparallele Rekonstruktionen. Zukünftige Arbeiten werden daher darin bestehen, Bestimmungen für Resttexturen einzuführen. In dieser Arbeit wird jedoch gezeigt, dass das vorgeschlagene Modell auch ohne diese Bestimmungen für Bitraten im niedrigen bis mittleren Bereich wettbewerbsfähig ist und gleichzeitig die für die zukünftige CC-VR erforderliche Funktionalität bietet. Darüber hinaus ist das Modell nicht darauf trainiert, die PSNR der Rekonstruktion zu maximieren, sondern die Wahrscheinlichkeit des Modells zu maximieren. Daher könnte die PSNR-Optimierung eine Möglichkeit sein, die RD-Leistung zu steigern, wie frühe Erkenntnisse zeigen. Schließlich müssen andere Eigenschaften und Anwendungen von SMoE untersucht und bewertet werden.

Es ist klar, dass es noch ein langer Weg ist, bis wir die ersten CC-VR-Anwendungen mit vollem Funktionsumfang nutzen können. Dennoch bietet diese Arbeit ein skalierbares Framework, das zukunftssicher ist, um den Weg zu einer vollständigen CC-VR zu erleichtern und gleichzeitig alternative und wettbewerbsfähige Methoden für die Darstellung und Codierung häufigerer Bildmodalitäten aufzuzeigen. Im Allgemeinen soll diese Arbeit in keiner Weise die endgültige Lösung sein, sondern eine neue Denkweise bei der Entwicklung von Bilddarstellungen eröffnen und einige potenziell veraltete Paradigmen aufgeben.

1 Introduction

1.1 Context

People have been eager to document and represent events and places in their lives for many years. The earliest evidence for this behavior are the first cave drawings (40,000 - 10,000 BCE). Many years later, during the 1800s, documenting became much easier with the introduction of analog photography. Photography further gave rise to analog cinema by capturing and displaying photographs in a fast-moving sequence. Nowadays, images and video mainly reside in the digital world and have never been as ubiquitous. The digitization of imaging further enabled digital manipulation of the images, ranging from photo editing to incorporating huge amounts of computer-generated imagery (CGI). Over the last decades, computer graphics have made enormous progress in terms of content-creation tools, rendering technology and hardware acceleration. These methods are not only applicable for CGI in cinema, but also allows for creating entire digital worlds. For example, gamers can roam across large areas and view objects far away or nearby with breathtaking quality. Even better, they can interact with the objects! These concepts would have been incredibly hard to imagine even fifty years ago.

Interestingly, the digitization of imaging and cinema did not change the way in which camera-captured visual data is experienced by the viewer. A movie director captures a specific viewpoint and then the viewers are shown what the director wanted them to see. However, the field of visual-storytelling has just started a new major shift. New hardware such as head-mounted *virtual reality* (VR) displays

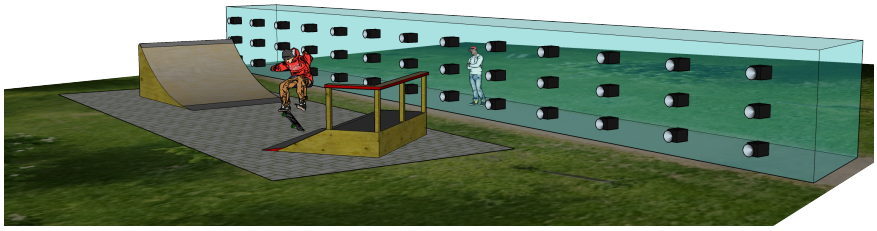


Figure 1.1: An illustration of a potential large-scale camera-setup for CC-VR 6-DoF applications (image courtesy: Niels Van Kets).

made people dream of achieving the same positional freedom for camera-captured content as they have in gaming environments. Imagine stepping into the story yourself with photorealistic quality: moving around and exploring the space behind objects. This would thus combine the positional freedom as in gaming with the photorealism of cinema. The only limitation would be that the viewer would remain a silent onlooker without the possibility to interact with the scene. Major events could then be captured in such a way that the viewer can relive the event from any point of view as a silent free-roaming spectator. Applications are not limited to entertainment or creating "living memories". It is easy to imagine applications in education, cultural heritage preservation, therapy, remote control in industry, medicine, real estate, tourism, remote presence, etc. However, capturing natural scenes will require a sufficient camera-coverage and will thus involve large camera setups. Fig. 1.1 and Fig. 1.2 illustrate potential camera-setups for CC-VR.

In traditional digital video, moving pictures are represented as a sequence of still images. These images (or "frames") thus have an explicit order, i.e. time. This logical ordering was the greatest source for reducing storage space for videos. Two consecutive frames tend to be very similar. Once one frame is known, then only the difference between that frame and the next frame needs to be stored or transmitted. However, these paradigms are challenged once the viewer is allowed to choose where to look at any time. There is no logical ordering anymore in the perceived frames. Furthermore, there exists a vast amount of viewpoints as a viewpoint can be located at every possible point in space and every possible gaze-orientation. This functionality heavily complicates the acquisition phase (camera setups), displaying technologies, and the bandwidth requirements. Such an increase of positional freedom thus pushes the requirements far beyond what is currently possible. Data compression and fast algorithmic navigation through these scenes will thus be of key importance in order to efficiently transfer and to display such data in real-time.

This dissertation explores a radically-novel method to enable this viewer-centric freedom of camera-captured content, without relying on CGI. The proposed method handles visual data in a way that efficiently allows the positional-

freedom application for camera-captured content. The method in this work leaves behind the paradigms that are present in traditional video coding used in industry standards such as JPEG and MPEG. The goal is to grab this shift in image modalities to leave behind all constraints presented in order to obtain a more abstract, more flexible, more future-proof method. For example, images and video have always had a one-to-one correspondence between capturing devices (light-sensitive grids), the storage model (pixel grids), and display device (e.g. grid of lights). Images were thus represented as a grid of colors, stored and displayed as such. In video, frames are taken at small intervals and stored as such. However, these 2-D pixels are only a sampling of the reality. In fact, light itself has no concept of image resolution or frames-per-second.

Capturing images and videos equals to integrating light spatially over small sensor elements and temporally based on shutter speed. Such an integration thus introduces a loss of information about the original data. Mathematically speaking, capturing images and videos equals to sampling an underlying continuous function that describes the traveling light in reality. The image is thus only a discrete sampling of a richer continuous and higher-dimensional function that characterizes the visual information i.e. the traveling light rays that surround us. The core intuition behind my work is that in order to represent the visual data in a scene, it is therefore more adequate to approach it in the same way, by storing an underlying pixel-generating continuous function that represents the total visual information in a scene.

Other recent developments have been impactful on the work in this dissertation. In the field of computer science novel tools have recently been developed for machine learning and pattern recognition. The applications are plentiful, e.g. object recognition, image segmentation, optical flow, motion estimation and also data compression. These recent developments in machine learning have given rise to powerful data-driven methods. In this work, many of machine learning concepts are employed without allowing ourselves to arrive at a magic black box, which is one common disadvantages of machine learning tools based on *artificial neural networks* (ANNs). Our aim is to obtain at a tractable, informative representation method.

1.2 Visual Immersive Experiences

The term “virtual reality” has become a much hyped and marketed buzzword and has thus left many confused about what exactly virtual reality is and what it is not. For clarity, a bird’s eye perspective of the recent developments is presented in this section while simultaneously introducing consistent terminology.

First of all, it is important to identify that *augmented reality* (AR) and virtual reality lay on a spectrum. VR is defined to be the extreme point on the spectrum

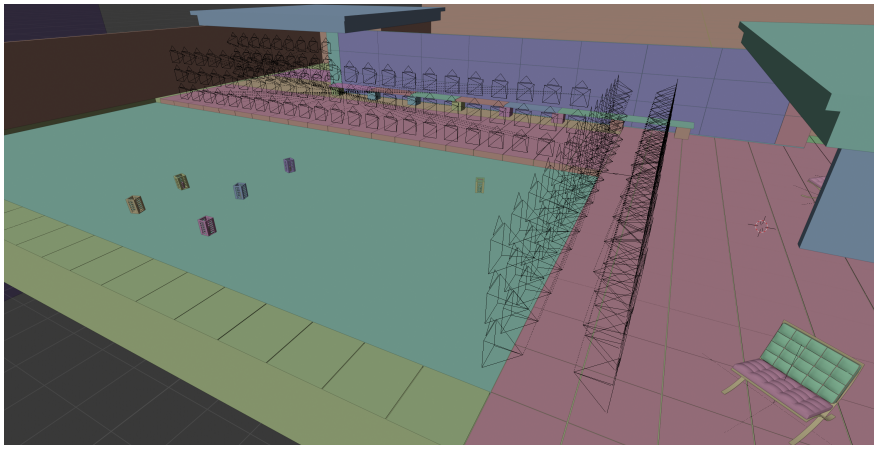


Figure 1.2: A schematic illustration of a CC-VR scene where viewers can walk around a pool. For example, the viewer can choose to see children playing in the pool, or to watch a conversation happening on the bench. The cameras are represented by small black wireframe pyramids (image courtesy: Martijn Courteaux).

where the digital images completely replaces everything the viewer sees. Then AR is the area of the spectrum where the viewer sees a mix of reality (see-through) and digital overlay images. Another differentiation that is important is where the overlaying digital imagery originates from. For example, the imagery could originate from computer-generated 3-D models with knowledge of surfaces, textures, material properties, etc; or the imagery could be camera captured. Exactly in the same way that a traditional movie playing on a household TV could result from cameras, e.g. a nature documentary, or could be entirely CGI generated, e.g. cartoons or CGI rendered films. Interestingly, the same continuum also exists from real to entirely fabricated as some content can be a mixture of both. Another distinction if the imagery is moving or not, i.e. contains a time dimension. This is typically indicated by the terms *static* and *dynamic*.

The difference of VR compared to traditional digital video is that the camera is not locked onto exactly one location in x, y, z-space and one gaze-direction. Whereas for immersive experiences, the subject enjoys higher levels of freedom and potentially is free to move in x, y, z-space and to look around. The level of freedom is typically expressed in an amount of *Degrees-of-Freedom* (DoF) and is expressed according to the following conventions.

3-DoF This typically describes 360-video in which the point of the camera is fixed. The number 3 refers to the three head rotations made possible, being roll, pitch, and yawn. The lack of any movement contributes heavily to the perceived motion sickness [1].

3-DoF+ This refers to having the same freedom as regular 3-DoF, with some extra limited translational freedom. Head movement in the (x, y, z) space is allowed, however, the movement is limited to a small sphere around the users' head. Consequently, a realistic immersive experience is possible for static end-users, e.g. viewers on a chair.

6-DoF This is the maximum of freedom in which a user can walk around, i.e. three translational movements, and three head rotations.

In general, virtual reality using CGI exclusively (CGI-VR) already allows for dynamic 6-DoF, and has done so since a long time. In 1992, the first 6-DoF 3-D game *Wolfenstein 3-D* was released [2]. This game is considered the "grandparent" of 3-D first-person shooter games which helped popularize the whole first-person experience in a virtual world. The main challenge is to achieve the same freedom for camera-captured content (CC-VR) without the knowledge of the exact 3-D scene.

Whereas images were a static snapshot from one viewing angle, video made the view dynamic. 360-video allowed the user to look around, but without allowing any translational movement (i.e. 3-DoF). The efficient representation and transmission of these lower-dimensional image modalities (e.g. images to 360-video) have been solved quite successfully. However, at this point the question remains on how to move towards to full 6-DoF which is currently a hot topic in industry and academia.

1.3 Towards 6-Degrees-of-Freedom CC-VR

In standardization, industry and academia, efforts are made in order to accommodate 6-DoF CC-VR. Two general strategies currently are being pursued for allowing true immersive experiences of camera-captured scenes: 3-D model construction and methods based on traditional video coding.

1.3.1 3-D reconstruction

The first strategy heavily relies on the exact geometry of the scene. In 3-D modeling, the scene is reconstructed in terms of 3-D objects with material properties using the camera-captured images. As such, the result is similar to the artificially created scenes as in gaming. Nevertheless, a 3-D scene construction based on images produces a reverse-engineering problem that is highly underdetermined. For example, complex material properties such as reflections need to be estimated and hole-filling methods are required in case of occlusions. Consequently, in order to arrive at a high fidelity reconstruction of the scene, a lot of prior information needs to be added. However, under constraints, this does perform well and provides easy

integration with existing CGI techniques, such as relighting the scene. As such, the 3-D reverse engineering approach does have the advantage of relying on optimized algorithms in 3-D graphics that have been researched and developed in the last decades.

Nevertheless, there are inherent difficulties of mimicking a natural scene. Most typical problems exist with all dynamic non-rigid objects with difficult diffuse reflection properties, e.g. water and smoke. Furthermore, the concept of "uncanny valley" plays an important role [4]. The uncanny valley states that when an artificial object mimics a human or animal, human observers tend to be very sensitive towards imperfections. These imperfections lead to a very unpleasant, uncomfortable perception. It thus requires a high level of detail in order to satisfy the viewer. This high level of detail then results in a larger bitrate and complex rendering steps.

An example is the Holoportation project, an end-to-end pipeline that allows the scanning, streaming and rendering of 3-D human meshes and objects [3]. The capturing is performed by employing several depth cameras facing inwards and using state-of-the-art methods for ensuring time-consistent meshes. However, the acquisition is limited to the acquisition of objects (outside-in capturing) in contrast to complete sceneries. Furthermore, the uncanny valley remains an issue as shown in Fig. 1.3.

In general, methods based on 3-D graphics easily enable the functional requirements for CC-VR. Nevertheless, the geometry and material estimation steps are computationally heavy, unnecessary, and lossy in terms of quality. Furthermore, one could argue that these steps are superfluous for the CC-VR use case as the user does not interact with the environment and no manipulation of the scenery is required. The proposed method in this work therefore focuses on not relying on the exact geometry of the scene, but to represent the light information in a physical volume as a whole.

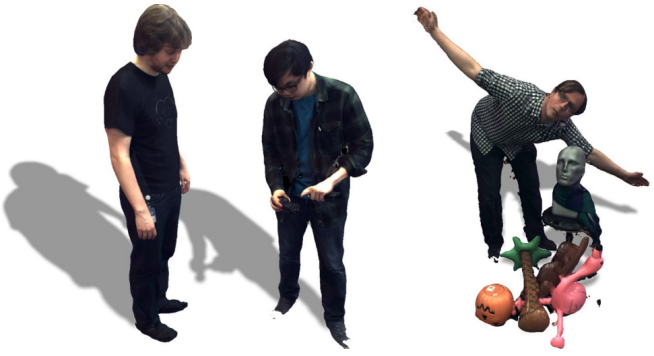


Figure 1.3: An example of 3D reconstructed humans and objects from the Holoportation project [3]

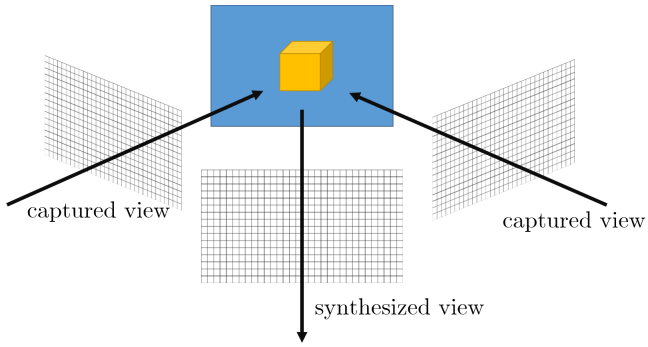


Figure 1.4: Schematic illustration of the basic idea behind a view synthesis process in image-based rendering.

1.3.2 Based on video coding

The second strategy relies on known hybrid transform/difference-coding techniques that are used in video, and, more recently, stereoscopic 3-D and 360° coding schemes. Following this philosophy, scenes are represented by coding a minimal set of 2-D images called "frames", and reconstructing the missing ones by view synthesis as illustrated in Fig. 1.4.

The biggest progress using this strategy is situated within standardization. The standardization organization *Moving Pictures Experts Group* (MPEG) launched an ad hoc group named "MPEG-i" for standardizing a codec for immersive video in three phases [5]. The first and second phase are aimed at respectively 3-DoF and 3-DoF+ immersive video. The final phase (MPEG-i Visual, starting in 2019) is aimed at a full 6-DoF immersive video. Currently, the call for test materials has been launched [6]. As such, no decisions have yet been made considering what representation will be used. The representations that are likely to be based on the 3-D extension of *High Efficiency Video Coding* (HEVC) or the successor of HEVC [5]. The 3-D extension of HEVC is based on *depth-image-based rendering* (DIBR) which combines video coding with view synthesis. The 3D-HEVC approach thus consists of two phases at the encoder side: (1) identifying a minimal set of representative views and corresponding depth maps and (2) compressing these views and depth maps [1]. The receiving side then performs some view interpolation using a view-synthesis process (as illustrated in Fig. 1.4) and hole-filling approaches. Note that depth maps suffer from some of the same drawbacks as 3-D graphics, in terms of non-rigid objects, time-consistency and details, e.g. hair and grass. Furthermore, it is possible that the geometry will be described by point-cloud coding instead of depth maps, which still has the same disadvantages as above.

In general, methods based on video coding provide excellent coding performance for video and for problems that can be translated to classical video. How-

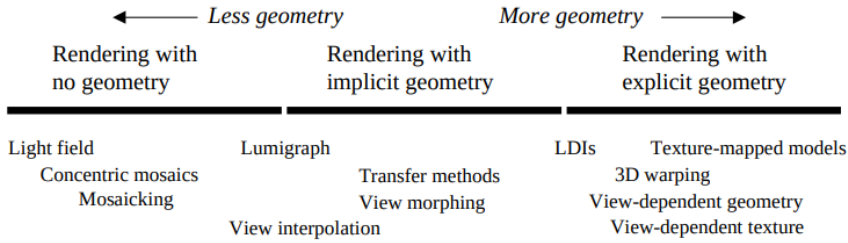


Figure 1.5: The scene-representation continuum [7]

ever, classical video coding is becoming less translatable (and thus less applicable) to VR, especially with future extensive user autonomy (high level of positional freedom). Firstly, the number of possible views grows exponentially with the level of user's autonomy. Secondly, the time-order in video is adequately exploited by motion-compensation, however, in VR, it is difficult to exploit the order of the frames as the end-user defines the order at playback. For example, the compression efficiency increases when more frames are predicted based on other frames. However, this also increases the number of interdependencies between the frames over space and time for motion and displacement compensation. Therefore, when a single view is requested by the viewer, a potentially large number of views to be transmitted and buffered at decoding side. Which can undo the obtained compression gain of the frame interprediction. Thirdly, the quality of the view-synthesis process is heavily dependent on the accuracy of the depth maps and inherently struggles with reflections, transparencies and volumetric effects, e.g. smoke. Finally, these systems do not cope well with irregularly-sampled data and heterogeneous camera setups in scene acquisition systems.

1.3.3 The scene-representation continuum

The two methods above describe two main classes of methods for storing and transmitting scene representations. In fact, hybrid methods that combine both strategies also exist and most methods can be put on a continuum between the two extremes: (1) purely 3-D modeling (meshes, point clouds, textures) and (2) extremely dense multi-view video coding without geometrical side-information. As mentioned, an example of such a hybrid technique is 3D-HEVC which uses depth maps to guide the view synthesis process [8]. These depth maps thus contain geometrical information, which then suffer from the same disadvantages of 3-D reconstruction. Similarly, in the work by Shum and Kang, several scene rendering methods are presented on a continuum [7]. Such rendering methods are not per se tuned towards the application of efficient data transfer or are meant as cod-

ing technologies. Fig. 1.5 illustrates several examples that range from no geometry necessary to pure 3-D graphics. These methods thus rely on certain representations that trade off the advantages and disadvantages of the two extreme representations.

The method on the left extreme of the spectrum is called *light fields* (LF). The light field corresponds to the total collection of light rays intersecting a plane or a volume. Light fields are an interesting technology as they do not rely on any geometrical information and provide a very useful theoretical framework. However, as will be discussed in Chapter 2, light fields have several practical disadvantages in terms of acquisition, processing and especially in terms of data efficiency. Nevertheless, light fields allow for photorealistic rendering of camera-captured content. Furthermore, it allows for refocusing and deriving viewpoints anywhere the light field provides sufficient information about the incoming rays at that point.

1.4 The Proposed Method

In this dissertation, a unifying method is proposed for representing digital imagery, ranging from images to video, light fields, and eventually full CC-VR image data. In essence, the observed 2-D views in a VR scene at each position and gaze orientation are 2-D slices of higher-dimensional light data. The proposed *Steered Mixture-of-Experts* (SMoE) representation is focused on the practical and information-rich distribution of such higher-dimensional visual data. Furthermore, the SMoE representation is made in a generic fashion that is applicable towards lower-dimensional image data such as images and video, as well as higher-dimensional imagery, e.g. 4-D light fields and eventually 6-DoF content. Additionally, the representation in this work is designed to follow the paradigm that lower-dimensional imagery is easily deducible from a higher-dimensional image model.

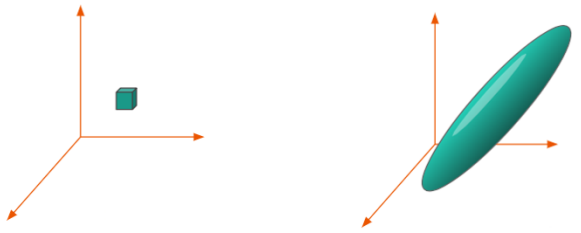
For image data with higher user autonomy, the proposed representation builds a statistical compact model of the light field, i.e. the light rays in a certain space and thus without explicitly relying on geometry in the representation itself. Note that the proposed technique therefore would thus lay on the left side of the spectrum in Fig. 1.5. As such, the disadvantages associated with geometrical imperfections are avoided. As shown later, the dimensionality of the data is dependent on the DoF level and higher autonomy typically involves higher-dimensional data. Consecutively, the increase of dimensionality leads to an increase of possible pixels. For example, as discussed in Chapter 2, light fields have a major disadvantage that they typically involve extreme amounts of data. Keeping the model compact will thus be one of the main challenges.

The SMoE method thus models the light information as higher-dimensional data, e.g. not as a set of 2-D images. More specifically, the method sparsely approximates the underlying pixel-generating function using inherently higher-

Figure 1.6: Illustration of the kernels in a SMoE model of the standard test image Lena*. Each kernel describes a region in the 2-D coordinate space and a gradient in the 3-D color space. Note that kernels have theoretical global support, however, the kernels are cut off for illustration purposes.*

dimensional atoms called 'kernels'. These kernels allow for simultaneous harvesting of pixel color correlation in various directions: e.g. time, pixel position, camera position. Fig. 1.6 illustrates the kernels for a 2-D image. These kernels should be viewed as a generalization of the pixel, i.e. the "pixel 2.0" as illustrated in Fig. 1.7. Not only do they have a spread along any number of dimensions, they also model gradients along these dimensions. Chapters 3 and 4 illustrate the workings of such kernels, and simultaneously provides an in-depth theoretical discussion of the model and its applications.

The rationale is thus that we should accept the fact that light data is intrinsically of high dimensionality, and we should not reduce the problem to a collection of 2-D problems as in video coding. The aim is to embrace the high-dimensional nature and to develop a tailored technique using inherently multi-dimensional image atoms. The representation was designed to have several number of desired properties that enable it to be a flexible and future-proof image representation. Those properties are elaborated in the following. Furthermore, a short discussion on how they relate to conventional image representations is provided.



(a) Pixel (b) Kernel

Figure 1.7: Illustration of a pixel versus a kernel in a 3-D coordinate space (e.g. 2D + time in video). In contrast to a pixel, a kernel defines a spread along each dimension and describes a gradient over all coordinate dimensions.

1.4.1 Compression efficiency

The first desired property of a representation is high bit-efficiency. This property has been the main focus for traditional video encoding since the beginning. Typically, the performance of encoding schemes is expressed in terms of rate-distortion (RD) performance, in which the image quality is measured in an objective or subjective manner in function of bitrate savings. The need for efficient compression remains important in the future. Especially considering that new image modalities produce extreme amounts of data. An example of one modern light field camera, the Lytro Immerge, produces 100 GB/s of raw data [9]. This camera captures an area with a horizontal span of roughly one meter. Even with the envisioned bandwidths of 5G [10], such data rates pose major challenges.

In contrast to traditional video coding techniques, the proposed representation is in fact a statistical model of the image pixel data. The mechanisms to introduce bitrate savings thus also go beyond the mechanisms in traditional video coding. Statistical models are typically used to provide a more tangible summarization of the underlying data. For illustration purposes, consider all the temperature information of the European Union since the 1960s. This surely amounts to a large body of data. A possible statistical model would then be, e.g. the mean temperature per month for the entire EU, or the average temperature per country. It is clear that these simple models would require a lot less storage than storing all the data. However, it is also clear that these models are not fine-grained enough for many applications and do not represent the underlying data well. On the other hand, a model that considers the average temperature per city while taking into account the temperatures of the last 30 days, would require in a considerable increase of model parameters to be stored, however, it would likely be more correct in providing estimations.

In SMoE, the bitrate reduction thus happens in two stages: (1) by restricting the number of parameters or the *model complexity* and (2) by quantizing the model parameters. First, a the statistical modeling provides a summarized representation of the underlying data. The model complexity can be tweaked to control the bitrate and consequently, the image quality. The goal is to have a model that is powerful enough to recreate the high visual quality towards the end user, while minimizing the model complexity. Secondly, the model parameters can further be coded with imprecisions in order to save bitrates through forms of quantization and entropy coding. Consequentially, the loss of image quality can thus happen in both of these stages. Note that the second stage does rely on techniques known in traditional video coding, e.g. coefficient quantization and entropy coding are still necessary but are performed on the model parameters instead of on transformation coefficients as in traditional video coding.

Nevertheless, mere bit-savings are not the primary goal of image and video formats any longer. For example, functionality that allows for flexible streaming of videos typically introduces a penalty in terms of bit-overhead in order to prevent re-encoding of data on the fly. Some examples of such functionality are: error-resilience, scalability in spatial resolution or quality, and random-access. Interestingly, one could also argue that some other external factors can influence the use of data formats that have less compression efficiency. For example, it has long been proven that the 25-year old JPEG is outperformed by the state of the art or even newer JPEG standards, e.g. JPEG-2000 [11]. However, the world-wide adoption of the standard proved to be more important than the bitrate reductions in newer formats [12]. In the following subsections, the desired extra functionality or properties of a representation are discussed.

1.4.2 Continuous representation

Traditional image and video representations rely on regular, dense sample grids. Such representations have thus had a one-on-one correspondence to the capturing or displaying hardware. An image was captured at a certain resolution on a regular grid, and then resampled in order to fit the output display. For video, frame resampling is done in order to match the desired output framerate. They thus contain sampled information from what is in fact continuous information in reality. Working with sampled data can lead to some cumbersome processing afterwards. For example, if a video is stored at a different resolution and a different frame-rate than a certain television can display, the frames need to be resampled at display-time.

Such fixed sample grids become more problematic in the light of 6-DoF. The scene is possibly captured by heterogeneous sets of cameras at irregular positions, at different spatial resolutions and frame rates. Consequently, the acquired pixel data is likely to be irregularly positioned due to such novel complex acquisition

systems [13]. Moreover, the scene can be displayed on different output devices simultaneous, each showing the scene at a different angle on possibly different resolutions. Storing the data as grids of samples is thus difficult if the combined set of captured samples is not necessarily on a grid.

Therefore, the presented representation is built to be a sparse continuous representation of the underlying function that gives rise to these samples. As such, no assumptions are made on the acquisition or displaying side. View reconstruction consists of merely sampling the continuous representation at an arbitrary resolution without the need for explicit view synthesis systems.

1.4.3 Random access

In conventional video, frames possess a natural order. A frame at time $t + 1$ is viewed after the frame at time t. In 6-DoF applications, this order remains for the time dimension, there is however no natural order in which the viewer will navigate through a scene. Traditional video coding techniques rely heavily on this order. Frames are predicted from the previous (or even next) neighboring frames. The temporal redundancy is minimized using motion compensation. If the same concepts want to be extended to 6-DoF then frames need to be interpolated from multiple frames nearby. However, these frames are now not distributed on a single time dimension, but scattered over space and time. It is therefore needed to buffer the reference frames and to engineer a multi-dimensional interpolation method.

The SMoE approach models the properties of the incoming light at a certain position. These properties are locally approximated by our multi-dimensional image atoms, i.e. the kernels. When a subject is at a certain point in space, only the local information in the close vicinity of the subject's head is needed. In other words, only the kernels that are responsible for this part of the scene need to be loaded. This can be easily cut out from the model as the information lies on the same real-world coordinate system. Note that this is inherently different when modeling a 3-D scene. In that case you need information of all possible 3-D objects in the line of sight. These objects could be at a large distance from the subject and that information is likely to be scattered over the disk. Also note that such granular random access is not trivially achieved using other methods from machine learning, e.g. in deep learning. This would require that you can cut out a subnetwork at all times that can be inferred independently.

1.4.4 Descriptive model

Machine learning approaches have started to dominate in image processing tasks such as object detection, face recognition, and many others. These approaches often rely on segmentations, intensity flows, depth information and others. Much of the digital imagery that is being produced now is made for machines to consume,

Figure 1.8: 3-D model and texture of a person's head used for model-based coding (Source: P. Eisert, FhG HHI, Berlin)

instead of humans. Conventional image representations rely on non-informative pixel samples on grids. These pixel grids thus first need to be processed in order to be understood. These methods are classified as being "blind". More recently, convolutional neural networks (CNNs) operate with impressive performance directly on the dense sample grids and learn the image features themselves. However, it is not trivial to scale CNN approaches to higher-dimensional image data.

On the other hand, coding methods have been researched that have a high understanding of the scene, i.e. in model-based coding approaches. However interesting, they were not very successful. The head-and-shoulders model for teleconference video services is a well-known example [14]. A generic 3-D model of a human's head is known at both encoder and decoder side and only the differences to such a model need to be transmitted. Fig. 1.8 illustrates a more recent 3-D model with corresponding texture. These methods produce a high-level of understanding of the scene that could be used for many post-processing tasks, but thus rely on many assumptions, e.g. knowledge about the content of the video.

Our belief is that the model should contain rather low-level descriptive features of the content without posing any assumptions on the contents, analogously to the MPEG-7 efforts [15]. Due to the data-driven approach of our method, the model itself takes on the form of the data. As presented later in Chapters 3 and 4, our method inherently provides multi-dimensional descriptive information about the segmentation, edges, intensity flow, and even depth in the case of 4-D light fields.

1.4.5 Pixel-parallel and light-weight decoding

Generally in video coding, there is the trend of highly increasing encoder complexity paired with moderate to low decoding complexity [16]. This is clearly desired in applications where the data is encoded only once, but decoded many times, e.g. *video-on-demand* (VOD) services. This type of scenario is the focus of this dissertation. Therefore, the decoder is aimed to be as light as possible. However, the encoding step is challenging in terms of computational complexity. In Chapter 5, it is discussed how to exploit the structure in the data to still obtain reasonable

encoding complexity.

The serial nature of the traditional video-coding paradigms (e.g. intra-prediction, motion-compensation) makes it impossible to really achieve pixel-level parallelism. Fine grained parallelism is becoming more and more desirable in algorithms as modern hardware tends to increase the number of execution threads rather than the speed of those threads. Despite the serial nature of video coding standards such as HEVC, parallelism in decoding/rendering is still pursued. HEVC relies on smart implementations, e.g. using a wavefront approach [17]. This ensures that blocks are decoded as soon as their dependencies are available. However, using 64-by-64 CTU blocks in a 1080p video only allows for 15 decoding blocks. In the case of 32-by-32 CTU blocks, one can achieve 30 parallel streams. Such a scheme does fit multi-threading architectures, but is less suited for massively parallel architectures.

In SMoE, decoding and image reconstruction are two separate processes. First, the quantized and entropy-coded model parameters need to be decoded from the bitstream. Secondly, from these model parameters, views can be derived/rendered. During rendering, pixels can be independently reconstructed at any desired resolution. A single pixel only relies on a limited set of kernels in the pixel's vicinity. Therefore, massively parallel implementations for rendering SMoE models are made possible, as shown in the work by Avramelos et al. [18].

1.5 Conclusion

The transition from traditional video to camera-captured 6-DoF applications is not trivial for a number of reasons. The main shift is that the added level of freedom gives rise to exponentially many possible viewing experiences, which all need to be covered by the representation. Traditional video coding methods rely on paradigms that were developed before such functionality was required and relied heavily on some properties that are not valid anymore, such as the known time-sequential order of the viewpoints. Methods that originate from 3-D graphics also are not trivially applicable as reconstructing a scene's geometry and material properties is a highly underdetermined problem. Therefore, in this dissertation, a novel method is proposed that is designed in a future-proof manner taking into account the desired functionalities for 6-DoF consumption.

Desired properties for a future-proof 6-DoF representation and coding scheme were identified and discussed. First, the method should provide high compression efficiency, although that the enabled functionality becomes of greater importance in 6-DoF applications. Secondly, a sparse continuous representation avoids problems related to sampled data, e.g. resampling, interpolation, etc. Thirdly, random access is of high importance as it is only known at view-time what parts of the representation are relevant for the viewer. Finally, pixel-parallel and light-weight

decoding is desired as this corresponds to the current evolution in graphics hardware.

The proposed Steered Mixture-of-Experts method is a sparse, continuous, statistical model that relies on multi-dimensional image atoms, i.e. kernels. These kernels should be seen as the generalization of the well-known pixel, i.e. a pixel that has a volume that is spread along all dimensions in the coordinate space of the image modality. Furthermore, a kernel does not represent a single color, but is in fact a linear function in terms of the coordinates which gives rise to multi-dimensional gradients along the coordinate dimensions. The proposed representation is designed in a generic fashion that is remarkably agnostic to the dimensionality of the image modality.

1.6 Outline

The dissertation is structured as follows. First, Chapter 2 introduces the mathematical foundations of light and the light field representation. The goal is to provide the reader with a clear understanding of the current state and challenges of light field processing. Secondly, the motivation behind this work is extensively discussed in the Chapter 3 as well as the proposed Steered Mixture-of-Experts method itself. The mathematical foundation is discussed in-depth. Furthermore, the model is illustrated on 2-D images which visually helps to understand the method. Additionally, several experiments are provided that validate early design decisions. Third, the SMoE method is applied to immersive image modalities in Chapter 4, i.e. 360-degree images, light field images and light field video.

Fourth, an important secondary contribution of this dissertation is presented in Chapter 5. The employed modeling algorithms are known to scale badly towards large amounts of data. However, immersive image modalities yield extremely large amounts of data. Nevertheless, it is shown in Chapter 5 that efficient implementations are possible by exploiting the structure of the data. Fifth, the main application in focus, i.e. coding, is investigated in Chapter 6. A dimension-agnostic coding scheme that binarizes the model parameters is presented and experimentally validated on 2-D images, 4-D light field images and 5-D light field video. Finally, conclusions and future work are presented in Chapter 7.

1.7 Publications

1.7.1 Journal publications

1. V. Avramelos, R. Verhack, I. Saenen, G. Van Wallendael, B. Goossens, and P. Lambert, “Highly parallel steered mixture-of-experts rendering at pixel-

level for image and light field data," Journal for Real-Time Image Processing, Dec. 2018.

2. R. Verhack, T. Sikora, G. Van Wallendael, and P. Lambert, "Steered Mixture-of-Experts for Light Field Images and Video: Representation and Coding," IEEE Transactions on Multimedia, pp. 1–1, 2019.

1.7.2 Conference proceedings

1. R. Verhack, A. Krutz, P. Lambert, R. Van de Walle, and T. Sikora, "Lossy Image Coding in the Pixel Domain using a Sparse Steering Kernel Synthesis Approach," in IEEE International Conference on Image Processing (ICIP), 2014, pp. 4807–4811.

2. R. Verhack, L. Lange, P. Lambert, R. Van de Walle, and T. Sikora, "Lossless image compression based on Kernel Least Mean Squares," in 2015 Picture Coding Symposium (PCS), 2015, pp. 189–193.

3. R. Verhack, T. Sikora, L. Lange, G. Van Wallendael, and P. Lambert, "A universal image coding approach using sparse steered Mixture-of-Experts regression," in 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 2142–2146.

4. L. Lange, R. Verhack, and T. Sikora, "Video Representation and Coding Using a Sparse Steered Mixture-of-Experts Network," in Picture Coding Symposium (PCS), 2016.

5. R. Verhack, S. Van De Keer, G. Van Wallendael, T. Sikora, and P. Lambert, "Color Prediction in Image Coding using Steered Mixture-of-Experts," in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2017.

6. R. Verhack, T. Sikora, L. Lange, R. Jongebloed, G. Van Wallendael, and P. Lambert, "Steered mixture-of-experts for light field coding, depth estimation, and processing," in 2017 IEEE International Conference on Multimedia and Expo (ICME), 2017, pp. 1183–1188.

7. R. Jongebloed, R. Verhack, L. Lange, and T. Sikora, "Hierarchical Learning of Sparse Image Representations Using Steered Mixture-of-Experts," in 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2018, pp. 1–6.

8. R. Verhack, G. Van Wallendael, M. Courteaux, P. Lambert, and T. Sikora, "Progressive Modeling of Steered Mixture-of-Experts for Light Field Video Approximation," in Picture Coding Symposium '18, 2018.

9. I. Saenen, R. Verhack, V. Avramelos, G. Van Wallendael, and P. Lambert, "Hard Real-Time, Pixel-Parallel Rendering of Light Field Videos using Steered Mixture-of-Experts," in Picture Coding Symposium '18, 2018.

10. V. Avramelos, I. Saenen, R. Verhack, G. Van Wallendael, P. Lambert, and T. Sikora, "Steered mixture-of-experts for light field video coding," in Applications of Digital Image Processing XLI, 2018, p. 11.

11. R. Verhack, N. Madhu, G. Van Wallendael, P. Lambert, and T. Sikora, "Steered Mixture-of-Experts Approximation of Spherical Image Data," in 2018 26th European Signal Processing Conference (EUSIPCO), 2018, pp. 256–260.

1.7.3 Awards

1. Top 10% paper - IEEE International Conference on Image Processing, 2014

2. Highly Commended Student Paper - 31st Picture Coding Symposium, 2015

3. Google Faculty Research Award (awarded to Prof. Sikora) for the entire research on Steered Mixture-of-Experts, 2016

4. Best Student Paper Award - IEEE International Conference on Multimedia and Expo, 2017

5. Finalist Best Paper Award - IEEE International Conference on Multimedia and Expo, 2017

2

The Plenoptic Function and Light Fields

2.1 Introduction

In order to enable the CC-VR requirements, it is important to first understand the human perception of light and light itself. This chapter mainly contains material that is known in the literature. A digest is presented in order for the reader to understand the current challenges in the field and how the proposed method maps onto these challenges.

It is important to understand that human perception is more than just an image being projected onto the retina. The abundance of optical illusions is proof that what we perceive in our brain is not an exact representation of the light that comes in. Our neurological visual system acts as a post-processor on the stimuli that it receives from the eyes. It helps us to maintain a smooth, consistent view. Furthermore, it is important to know the limitations of our vision. Even though light rays have a very large color spectrum, humans can only perceive three relatively narrow subbands, corresponding to the three different cone-types in our eyes.

> Our goal is to provide a reproduction of the stimuli that eventually lead to images in our minds. Ideally, we search for the minimal amount of information necessary in order to reproduce the same level of perception in the brain.

In essence, the observed 2-D views in a VR scene at each position and gaze

orientation are 2-D slices of higher-dimensional light data. This 2-D slice is then sampled at a certain resolution. In any space, we are surrounded by a vast amount of light rays irradiated by light sources and bouncing back from objects. Our eyes integrate these light rays onto our retina which consequently produces a mental image. The distribution of light rays in a space is characterized by a concept called the *light field* (LF).

However, working in high dimensional spaces quickly becomes challenging. In general, all problems associated with working in higher dimensional spaces are grouped under the name "curse of dimensionality". The transition from images to video, i.e. going from 2-D to 3-D (2-D + time), has proven to be challenging in the past and has been the subject of decades of research. Moving even further in dimensionality to dynamic 5-D (4-D + time) light field videos thus poses many challenges to the community. Acquisition, processing, displaying, compressing, and transmission of light field data are currently hot topics in the image processing field. The adoption of light field cameras has given rise to new applications, ranging from their initial purpose (photorealistic image-based rendering [19]), to current computer vision applications that make use of their rich encoded information; these include 3-D reconstruction, segmentation and matting, saliency detection, object detection and recognition, tracking, and video stabilization [20].

The following two sections provides firstly a mathematical framework for us to work with in the coming chapters. Secondly, an overview is provided of the current state of the art and focus on challenges that are relevant to the proposed SMoE approach.

2.2 The Mathematics of Light

2.2.1 The 5-D Plenoptic Function

The plenoptic function is a mathematical representation of light rays in a space. It was first described by Adelson and Bergen and describes light as a 7-D function [21]. The plenoptic function describes a light ray arriving at position (x, y, z) in space and describes the intensity I of the ray arriving at angles (θ, ϕ) at wavelength λ at time t:

$$I = P(\theta, \phi, \lambda, t, x, y, z) \tag{2.1}$$

However, it is common in literature to simplify this to a 5-D function assuming a monochromatic and time-invariant function. Consequently, we arrive at the 5-D plenoptic function. The time-invariant assumption comes from the idea that the time dimension t can be recorded in different frames and that the wavelength can be coded in 3 different color channels [20].

Note that humans sample this function at two viewing positions at a time along a line in (x, y, z)-space axis, one location for each eye. The wavelength axis is sampled with only three cone types. The most densely sampled axes are those corresponding to the visual angles (θ, ϕ) which define the spatial resolution of our vision. Time is the only axis that humans sample continuously [21]. The first following steps in our visual "reasoning" then consist of examining the local properties of the plenoptic function, such as low order derivatives to identify edges [21].

2.2.2 The 4-D Light Field

Another simplification of this 5-D function can be made under some conditions. Note that light rays travel straight and do not interfere with each other. Therefore, one dimension is redundant in a space where no objects are present. In other words, if all light rays (and their incident angle) crossing a single 2-D plane are known, then the light rays that intersect a point in (x, y, z) space can be derived. Levoy and Hanrahan introduced this concept of the 4-D light field, which in the case of open space defines the full plenoptic function [19]. The parametrization comes in several forms. The most common representation is to consider the coordinates of the intersections of the light rays with two parallel planes at arbitrary distance. (u, v) denoting the coordinate on the first plane, (s, t) being the intersection of the second plane [20]. A specifically handy parametrization is when (u, v) is considered as the camera plane with their focal plane on the (s, t)-plane.

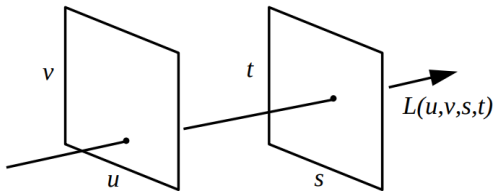


Figure 2.1: The 4-D light field representation shown in (u, v, s, t) parametrization [19]. Each light ray is identified by the intersections of the rays with two parallel planes.

In this work, the parametrization (a_1, a_2, x_1, x_2) is used in which (a_1, a_2) represent the camera coordinate and (x_1, x_2) is the coordinate of the pixel on the image sensor. The (a_1, a_2) plane is thus equivalent to the (u, v) plane, however the (x_1, x_2) is now relative to the camera coordinate instead of the absolute coordinates (s, t). In other words, the top left of a view always corresponds to origin, i.e. $(x_1, x_2) = (0, 0)$ This parametrization is very practical as it allows us to represent the 4-D light field as a 2-D matrix of 2-D images as shown in Fig. 2.2. This representation is common in practice, e.g. in the light field toolbox for MATLAB [23].

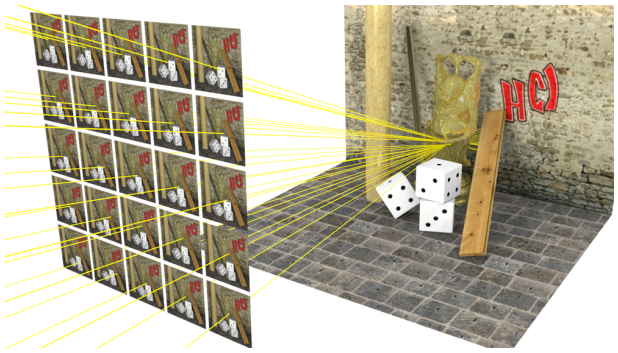


Figure 2.2: The light field as a 2-D matrix of views (Image courtesy: Heidelberg Collaboratory for Image Processing HCI).

As it is hard for humans to visualize 4-D data, a common practice is to visualize the 4-D light field as three 2-D slices of this 4-D space. The 2-D slice along the camera dimensions (i.e. camera-coordinates are fixed) corresponds to a view, also dubbed a sub-aperture image. The 2-D slices along one of the image sensor dimensions and a camera dimension corresponds to an *epipolar plane image* (EPI). The concept of EPIs is crucial in light field processing and is most easily understood in the form of camera movement. Fig. 2.3 illustrates a horizontally moving camera from right to left. Imagine stacking the frames of the video along the time dimension. The resulting stack of frames thus results in a 3-D spatio-temporal volume. The EPI corresponds to a slice (indicated as a dashed line) of the spatio-temporal volume. The EPI visualized thus corresponds to a single horizontal line of pixels and how these pixels change over time. In the EPI, it can be seen that in the beginning (left) only the third (green) and the second (orange) object is visible, whereas the green object disappears after some time and then the first (blue) object enters te view. From the beginning until almost at the end, the orange and furthest object remained visible. At the very end, all objects are out of the field of view. The resulting EPI is shown that is cut at a certain pixel row u. Note that the (a_1, a_2) camera plane can be seen as respectively vertical and horizontal camera movement.

Fig. 2.4 illustrates a visualization that shows the spatial views as a grid of images as well as two EPIs for the light field visualized in Fig. 2.2. Similarly, Fig. 2.5 illustrates the 4-D representation as three 2-D slices (including two EPIs) for a part of a camera-captured LF. The top left slice shows the two spatial dimensions and is thus a view at from a certain angle. Two EPIs are shown on the right and on the bottom and thus represent one spatial dimension on the image sensor, and one camera displacement dimension. One such EPI contains information in both spatial and angular dimensions [20]. In this particular image, these correspond with

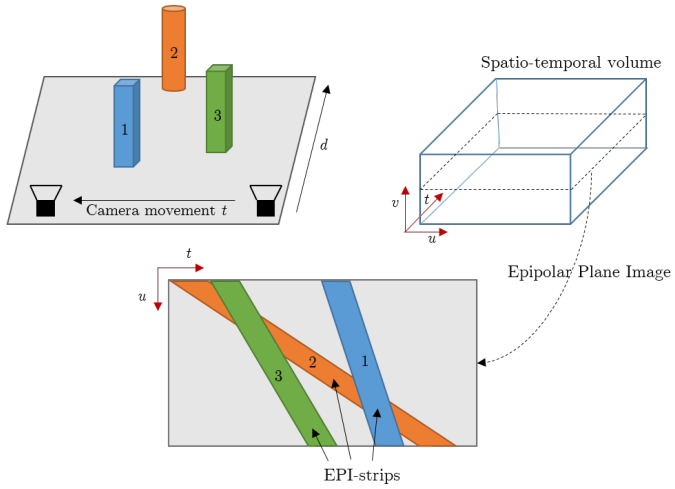


Figure 2.3: Visualization of the epipolar plane image (EPI) concept with a horizontally moving camera from right to left (adapted from [22]). The images captured on the (u, v) sensor plane form a cube when stacked over the time dimension, i.e. the spatio-temporal volume. The bottom image shows EPI, i.e. the 2-D slice of the volume along the dashed line. Note that the diagonal structure of the strips in the EPI and how the slope of the strips are indicators of the depth of the corresponding object.

how the red lines respectively change when the subject moves its head left-right (bottom EPI) or top-down (right EPI).

The 4-D light field proves to be practical, especially under the idealized Lambertian reflection assumption. This assumption states that the perceived intensity of the set of light rays emitted by a given source point is independent of the viewers location. Consequently, the lines of the same intensity (and color) can be followed in the EPIs. In the particular case of Fig. 2.5, it can be seen that the white background in the center view clearly is identifiable in the EPIs. This has interesting applications for depth estimation. Points with different depths can be visualized as lines with different slopes in the EPI. Conversely, the slopes of the lines in the EPI reflect the depth of the scene captured by the light field [20]. It is worth noting that the two planes can be replaced by a sphere, resulting in 360° or omnidirectional light fields.

2.2.3 Viewport rendering

In order to generate a view, the light field merely needs to be resampled. As such, light field rendering can be efficiently implemented [19]. Fig. 2.6 illustrates the main principle of light field rendering. Rendering viewpoints on the camera plane is trivial. On integral displacements, the views correspond to the captured views.

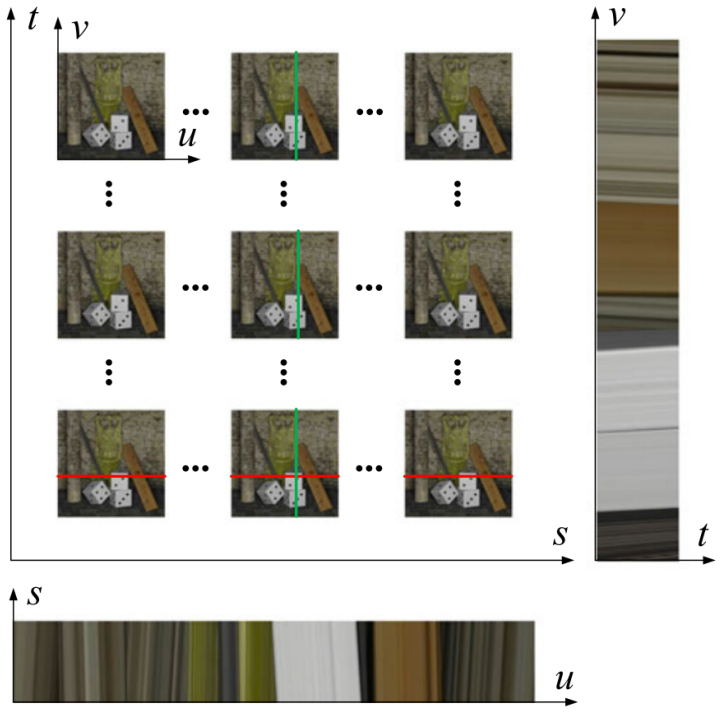


Figure 2.4: This visualization simultaneously shows the spatial as well as the angular dimensions of the light field in Fig. 2.2 [20]. The views are shown as a matrix of 2-D images, whereas each view has the (u, v) spatial coordinates and are arranged in a raster by the angular (s, t) coordinates. On the bottom and to the right, two EPIs are shown based on the green and red spatial cuts. An EPI thus contains both a spatial and an angular dimension.

At non-integral positions, i.e. between captured views, simple linear interpolation between neighboring views is sufficient when the views are captured densely enough. If a view is to be rendered outside of the camera plane (stepping forward or backwards), then pixels from several views are combined. Which pixels from which views is determined by projecting the new virtual camera sensor's pixels onto the focal plane and back-projecting the intersections of the focal plane back onto the original captured views. From a practical point of view, stepping out of the camera plane is computationally more challenging as it potentially requires considerable memory for storing all views. Furthermore, the pixel projection requires careful implementation to be efficient. Interestingly, the novel views can be refocused on a desired virtual focal plane and using an arbitrarily depth of field by using various strategies [24]. This allows the valuable application of refocusing in post-production.

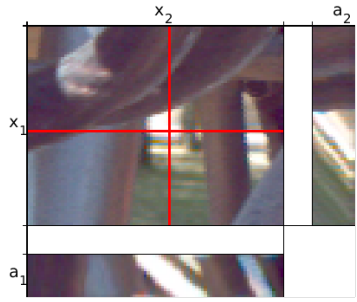
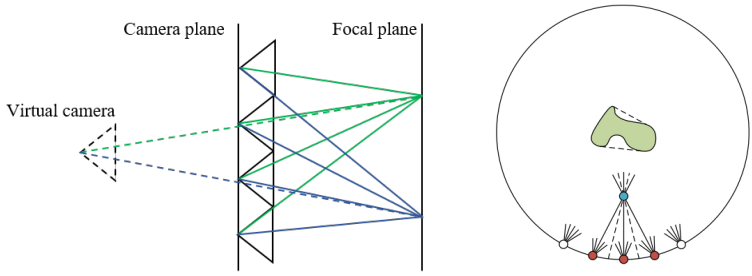


Figure 2.5: Three 2-D slices of a camera-captured 4-D light field using the parametrization of this dissertation. The top left slice shows the two spatial dimensions corresponding to a view from a certain view angle. EPIs are shown on the right and on the bottom, the 2-D slices have one spatial and one camera dimension. In this particular image, these correspond with how the red lines respectively change when the subject moves its head vertically (bottom EPI) or horizontally (right EPI)



(a) Planar (b) Circular [20]

Figure 2.6: Illustration of the main principle behind light field rendering. When rendering a virtual viewpoint (dashed camera), pixels from captured viewpoints are mixed together to form the new virtual viewpoint. In order to do so, the relevant light rays are backprojected onto the original camera viewpoints or interpolated from multiple viewpoints if necessary. The rendering of a virtual viewpoint behind a camera plane is shown in (a), whereas a viewpoint that is within the ring of cameras, i.e. closer to the object is shown in (b).

2.2.4 Images and video as functions

Images and video are lower-dimensional slices of the plenoptic function, in the same way a circle is a 2-dimensional slice of a 3-D sphere. In general, in this dissertation, the goals is to preserve the property that lower-dimensional images can directly be sliced from higher-dimensional image models. In order to do so, this section firstly shows how known image modalities, such as images and video can be expressed as functions.

A captured image can be seen as a sampling of an underlying image function

which has a 2-D coordinate space (the location of the sample) and a 1-D color space for grayscale images, or a 3-D color space (e.g. RGB) in the case for color images. For consistency, the coordinate space is referred to as X and the color space as Z.

$$f : X \rightarrow Z : (x_1, x_2) \rightarrow (R, G, B) \tag{2.2}$$

Analogously, video can similarly be seen as a sampling of a spatio-temporal 3-D coordinate space (2-D sample location, 1-D time dimension). The color space remains the same as for images.

$$f : X \rightarrow Z : (x_1, x_2, t) \rightarrow (R, G, B) \tag{2.3}$$

Consequently, the 4-D light field introduced above can be seen as a function having a 4-D dimensional coordinate space. Two dimensions (x_1, x_2) indicating the sample location inside a view, and two dimensions (a_1, a_2) that indicate which view on the 2-D camera grid.

$$f : X \rightarrow Z : (a_1, a_2, x_1, x_2) \rightarrow (R, G, B) \tag{2.4}$$

In practice, the color space $YCbCr$ is more commonly used in image processing as it separates the luminance Y (brightness) from the chrominance (Cb, Cr) (color information).

2.2.5 What to use for CC-VR?

The problem with the 4-D light field is that the assumption of the light being in free space does not necessarily hold for CC-VR. Although the 4-D light field might not be perfectly suited for CC-VR applications, it does provide a tangible step up towards the full plenoptic function and a powerful proof of some of the core strengths of the proposed method. Theoretical and experimental results are presented in Chapter 4. Moreover, the principles of light fields are extremely valuable during the acquisition of 6-DoF content as the open-space assumption always holds locally, i.e. over a smaller subspace of the entire walkable space for the viewer. An open space requires that no object is in the navigatable space. However, one could divide up a space into a set of smaller open spaces.

The idea behind this dissertation is that 6-DoF scenes are continuous-time variant and that the color channels $YCbCr$ should be included in the parametrization as 3-D vector. By modeling the color channels simultaneously, it becomes possible to capture potential inter-channel correlation. Therefore, in this work, the following parametrization is proposed for 6-DoF digital content:

$$f : X \rightarrow Z : (x, y, z, \theta, \phi, t) \rightarrow (Y, Cb, Cr), \tag{2.5}$$

with X being the 6-dimensional coordinate space and Z being the 3 dimensional color space. The fact that the coordinate space is 6-D for 6-DoF is coincidental, as the six in 6-DoF refers to six distinct movements that are possible by the viewer. Note that this parametrization is similar to the ray-space representation by Tanimoto for *free-viewpoint television* (FTV) [25]. FTV was the predecessor for MPEG-i which was similarly based on view synthesis. In Chapter 6, FTV and the ray-space representation are further discussed in comparison to the proposed SMoE coding approach.

2.3 Current Light Field Processing

In this section, an overview is provided of the current relevant state of the art of light field capturing and processing. Wu et al. published an exhaustive overview on which this section is based [20]. The state of the art is mainly focused on the 4-D light field framework where the cameras are structured on a plane or a sphere as introduced above. Similarly, the main experimental validation of the proposed theory in Chapter 4 is also performed on such 4-D image data.

Fig. 2.7 illustrates a taxonomy and the interplay of typical processing aspects in light fields [20]. The classification distinguishes between low-level, mid-level and high-level processing. Interestingly, several of the mid-level processing tasks rely on each other, therefore it can be argued that it is beneficial to provide a single representation for all of these tasks. As such, the proposed SMoE representation is intended as a representation adequate for mid-level and high-level processing. Light field displaying can be done by showing the rendered 2-D views on conventional hardware, VR headsets, or dedicated light field displays that can irradiate a number of views simultaneously. In this dissertation, there is no assumption made on displaying technology and any displaying technology is expected to be supported. For more details on light field displaying, the reader is referred to the overview paper of Wu et al. [20]. The rest of this section will elaborate on light field capturing/acquisition and mid-level processing tasks consecutively that are related to the proposed representation.

2.3.1 Light field acquisition

Light fields describe the distribution of light rays in a 3-D space. This is unlike conventional images, which record the 2-D slices of the light rays by angularly integrating the rays at each pixel [20]. It is thus easy to extract a 2-D image from a light field. However, the challenge lays in capturing the light field. Current capturing hardware is limited to 2-D image sensors. These sensors sample the light field. Given enough 2-D images, the full light field can be approximated.

There are currently three main methods of capturing light fields: multi-sensor

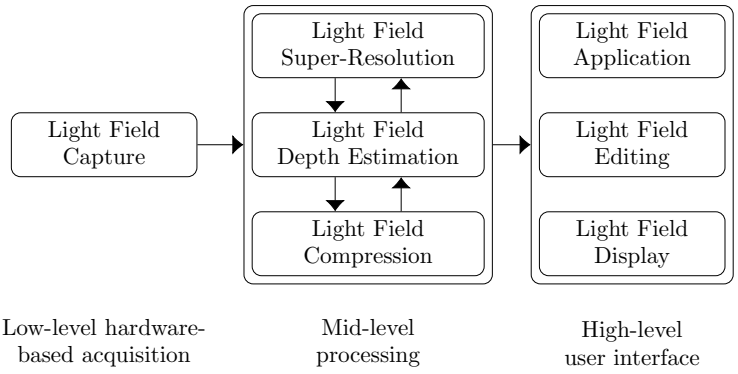


Figure 2.7: Overview of light field processing [20].

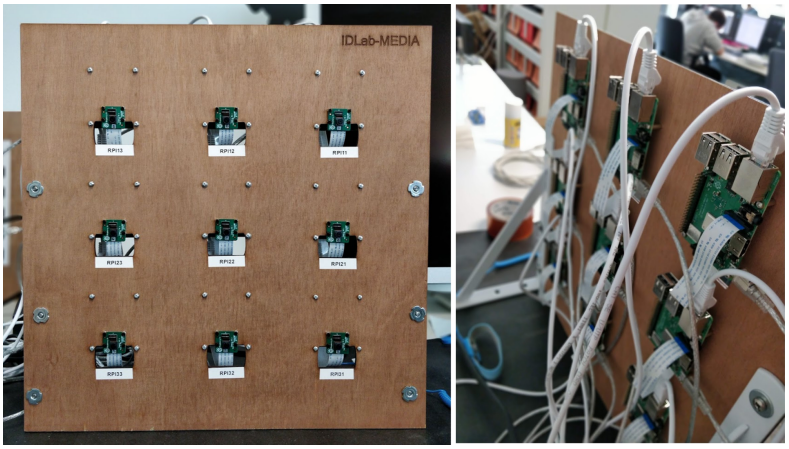


Figure 2.8: Low-cost light field camera array built by the author during his PhD. The array consists of a 3-by-3 matrix of RaspBerry Pi computing modules (right) with attached camera (left).

capture, time-sequential capture, and multiplexed imaging. The multi-sensor capture approach requires an array of image sensors distributed on a plane or on a sphere that capture light field samples from different viewpoints. These arrays can thus be build using a set of conventional image cameras. This approach is able to capture a light field instantaneously and is thus competent to record light field sequences (light field video). Early multi-sensor systems were bulky and expensive, however, recently cheaper and more portable designs have increased the potential of this acquisition method [20]. Fig. 2.8 shows a light field camera array that I constructed consisting of 3x3 camera array built with low-cost RaspBerry Pi 3 (B+) computing modules each with a RaspBerry Pi Camera V2 image sensor.

Figure 2.9: Two rotating camera setups developed by Google that work using time-multiplexing in combination with a multi-camera setup to obtain spherical light fields [27].

Secondly, time-sequential capture systems use a single sensor that moves around the scene, capturing views from different angles [19]. These methods require high precision of the movements of the sensor. The main advantage is that only a single sensor is required, however, the capture process is by definition time consuming and thus can only be applied to static scenes. Fig. 2.9 illustrates combinations of a multi-camera array combined with time-multiplexing in order to achieve spherical light fields.

Finally, multiplexed imaging aims to encode the 4-D light field into a 2-D sensor plane by multiplexing the angular domain into the spatial (or frequency) domain. It thus allows to sample the entire 4-D light field, but imposes a trade-off between spatial and angular resolutions. The most well-known method is spatial mutiplexing, such as used in the commercially available Lytro Illum camera [26]. These so called *lenslet*-based cameras, have an array of microlenses on top of the 2-D image sensor and thus provides a grid of views with very low displacement. Fig. 2.10 illustrates a lenslet image resulting from a Lytro Illum camera. This 2-D image thus contains both spatial and angular information. The area under a single microlens is referred to as a macropixel and contains all angular information for a single pixel. By gathering pixels in the same coordinate of each macropixel, an image located at a certain viewpoint can be obtained [20].

The views need to be captured sufficiently dense in order to render views without ghosting effects. More precisely, the maximum disparity between two corresponding pixels in neighboring views must be less than 1 pixel, a value closely related to the camera resolution and scene depth [28], [29]. In general, the quality of the interpolated points increases when the disparity decreases. The need for adequate sampling is thus one of the pitfalls in light field rendering. To mitigate this problem during acquisition, the light field can be inferred by irregularly sampling the scene, combined with geometrical information [30]. Note that, once the

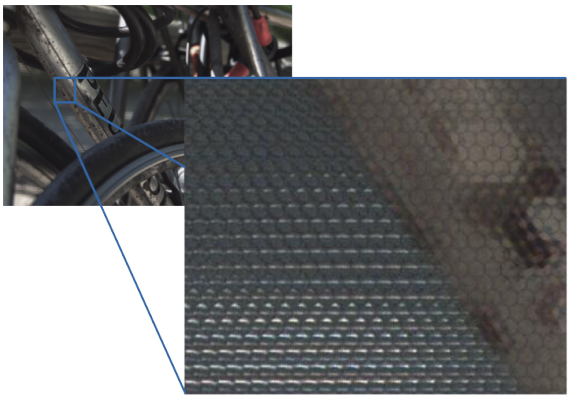


Figure 2.10: Example of a lenslet image resulting from a Lytro camera [26]. Each macropixel holds the color information about one pixel-location in a view for a multitude of viewing angles.

light field is inferred, the rendering is performed without the use of the geometrical information.

Wu et al. predict that following the current trend of miniaturization and maturation, light field cameras could find its way into mobile devices, such as smartphones and tablet computers, in the near future [20].

2.3.2 Light field super-resolution

The rich information embedded in the light field allows for numerous image processing tasks. Firstly, super-resolution is possible in both spatial and angular domains. Spatial super-resolution allows us to arrive at views of higher resolution compared to the resolution of the cameras used. This is possible by the multiple exposures of the same scene that are at nonintegral displacements. Pixels in neighboring views that are not on exactly the same grid, can be propagated to the target view [20]. Angular super-resolution or view-interpolation allows us to synthesize views in between captured views. A very effective, although elaborate view synthesis for wide-baseline camera arrays was introduced in [31]. More recently, a view-consistent spatial super-resolution method was proposed using low-rank approximation of the angular dimension combined with CNN restoration [32].

In the proposed method in the coming chapters, pixels are not assumed to be at integral positions when building the model. Furthermore, the model builds a continuous representation of the entire 4-D space. This technically enables inherent super-resolution. Nevertheless, the exhaustive assessment of the efficacy of the proposed research and pixel-based methods is considered future work.

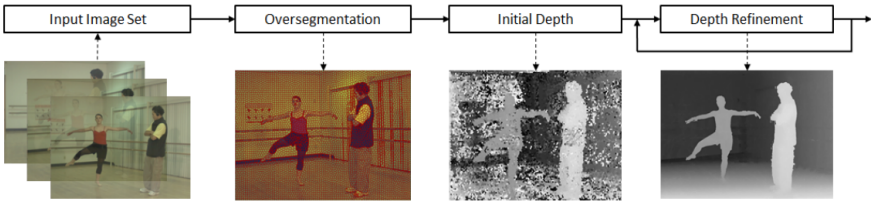


Figure 2.11: Example of depth estimation based on super-pixel segmentation with an initial depth estimation and an iterative depth refinement stage [33].

2.3.3 Light field depth estimation

Light fields allow to estimate the depth of the scene very precisely as a large set of dense views of the same scene is available. As illustrated in Fig. 2.11, estimating the depth typically involves two stages: (1) an initial depth estimation and (2) a refinement stage [20]. First, the initial depth estimation is performed using either (a) match-based, (b) EPI-based or (c) learning-based methods. In (a) match-based methods, the depth is estimated using techniques from stereo-matching where features are mapped in adjacent views. The method (b) is the most interesting in the context of this work as it is linked to the implicit raw depth estimation that is possible using the proposed framework in Sec. 4.3.2. These methods analyze the EPI as illustrated in Fig. 2.3. As mentioned above, the slopes of the EPI strips correspond to the depth of that object in space. The (c) learning-based methods rely on various machine-learning methods e.g., using CNNs to predict depth information.

Due to noise, occlusions or inherent matching uncertainty caused by textureless regions, the initial depth map typically contains outliers [20]. Therefore, the second refinement stage is employed to smooth out outliers in the depth map. Typical methods are either within a Markov random field framework or a variational framework [20].

2.3.4 Light field compression

The acquisition of light fields produces enormous amounts of data. Imagine a camera array of 250 horizontal and 50 vertical full HD cameras with a baseline (disparity between cameras) of 0.5cm. This produces a camera plane with a support of 125cm x 25cm. However, this requires 250x50=12.500 images of 1920×1080 pixels. In the case of light field video at 60 frames-per-second, this results in 750.000 full HD images or 1.5×10^{12} pixels per second or 3.6×10^{13} bits (4.190 GB/s) at 8-bit per color channel! Luckily, there is tremendous redundancy as the views capture the same scene with only slight camera shifts. It is therefore no surprise that light field compression is currently a hot topic. In the following, an overview is provided in which approaches are grouped as follows: (1) based on

view prediction strategies, (2) based on 4-D transform coding, (3) based on neural networks, and (4) segmentation-based. In general, coding methods can be divided into methods that operate on the lenslet sensor image and methods that operate on the 2-D stack of 2-D images (cfr. multi-sensor capturing). The former thus has limited applicability as it is mainly limited to light fields coming from these specific lenslet hardware architectures, whereas the latter is more general.

Firstly, most proposed light field coding approaches are based on prediction schemes that use existing video coding tools under the hood or extend these methods. In the case of lenslet images, intra-prediction can be applied directly on the lenslet sensor image. Examples include self-similarity [34] and local linear embedding [35] intra-prediction methods, both embedded into HEVC. However, these methods are only applicable for these specific lenslet hardware architectures. A more hardware-agnostic approach handles the light field more generally as a 2-D matrix of 2-D camera views. Such coding schemes often rely on video coding techniques by forming a pseudo video-sequence of the captured views that serves as input for HEVC [36]. Pseudo-video coding is commonly used as an anchor in light field coding [37], and is also used as such in the light field coding experiments in Sec. 6.5. Multiple improvements over the same basic idea have been proposed that enable some extra functionality or further coding gains. For example, a hierarchical reference structure was proposed for inter-coding of the pseudo video-sequence [38]. Similarly, a coding method that allows for field-of-view scalability was proposed using HEVC as a base layer combined with an exemplar-based inter-layer prediction [39]. Instead of ordering the views temporally, a method based on the multiview extension of HEVC (MV-HEVC) has also been proposed [40]. Similar to the extensions on the pseudo-temporal video coding, extensions were proposed to add desired functionality. For example, a hierarchical multiview structure was also proposed to improve the random access functionality and parallel processing encoding capabilities [41]. The advantage of using the MV-HEVC structure is that light field video can thus also be supported. Avramelos et al. evaluated several prediction structures in terms of coding efficiency and random access functionality for light field video [42]. Based on these findings, they proposed a prediction scheme that balances compression efficiency and adequate random access granularity. In their proposal, each center-view is predicted temporally from the previous time instance. All other views are then predicted from that middle view.

The above methods can be further optimized by reducing the number of views to be sent by selecting the most important views and reconstructing the other views at decoder side, similar to 3D-HEVC and MPEG-i. One example is the compression of lenslet light fields using structural key views [43], in which ideas of compressed sensing are incorporated in order to achieve minimal parameters to define the whole light field. Similarly, a scheme was proposed where the key views are

encoded using MV-HEVC and then other views are reconstructed at decoder side using a CNN [44].

Secondly, 4-D extensions of well-known transforms have been applied to light field coding. For example, the standardization organization *Joint Photographic Experts Group* or JPEG started standardization efforts for coding methods targeting light fields. This is as part of their larger ambition of JPEG-Pleno, which is aimed to arrive at a single unifying format for point-clouds, light fields and holographic image data [45]. One promising method under consideration is a 4-D DCT-based codec [46]. This method achieves very competitive results on lenslet LFs using a conceptually simple design. However, it remains a dense representation that requires regularly sampled data, and also requires as many coefficients in memory as there are pixels in all views. Furthermore, the efficiency on wide-baseline light fields is not ensured as larger shifts in views introduce discontinuities along camera dimensions and discontinuities are usually not well represented by a DCT. Similarly, the 4-D wavelet transform has been investigated for light field compression [47].

Thirdly, interesting work on light field compression is also coming from the field of machine learning for both lenslet images and 4-D light fields. For example, Schiopu and Munteanu proposed a CNN-based method for lenslet images where macropixels are predicted based on the surrounding macropixels for lossless light field coding [48]. Alternatively, Alperovich et al. proposes a deep encoder-decoder network for 4-D light field patches [49]. The proposal follows a fully convolutional autoencoder architecture in order to reduce the complex light field into a low-dimensional representation, which is suited for coding. Most interesting is that they jointly learn to discriminate relevant features of the light field. The decoder can decode the patch into diffuse and specular components or return a depth map for the center view. This is an example of a feature-rich representation similar to the proposed method in this dissertation, which also can be used to calculate the depth maps for each view (see Chapter 4).

Finally, segmentation-based methods have also been presented for light field coding. Rizkallah et al. proposed work that utilizes coherent super-pixels over-segmentation of the views in combination with graph-based transforms in order to capture correlation in the LFs [50]. Whereas, in other work, light fields are segmented into “super-rays”, i.e. 4-D clusters of pixel data that correspond over all views [51]. Given such a segmentation, coding methods can be devised, e.g. based on a 4-D shape-adaptive DCT [52] or a singular value decomposition [53]. These methods bare similarity with the method proposed in this dissertation, as the proposed method implicitly provides a segmentation of the pixel data.

2.4 Conclusion

The theory of light fields was developed over twenty years ago but is currently undergoing a revival. The recent rise in attention is likely driven by the development of improved acquisition systems and increased computational power. It is clear that light fields contain extremely informative descriptions of the scene. Furthermore, the rendering applications allow 6-DoF for viewers and even allows for refocusing at render-time. The implicit presence of depth allows to perform depth-based filtering, which enables post-processing without using green screens. The possible applications of light fields are numerous, but the current bottleneck lays in acquisition and the processing of enormous amounts of samples. Compression is therefore crucial and furthermore, coding schemes for light fields are now challenged by the versatile applications of light fields. In video, only sequential frames were displayed. Now, samples from all possible views can be queried at display time when performing light field rendering. Fast pixel-independent sampling from all views is thus crucial for real-time rendering. Nevertheless, in research and standardization, there is a strong inclination to continue to work with classical paradigms from video coding that might need to be revisited.

The premise of this dissertation is to explore a method that represents such inherently high-dimensional data using inherently multi-dimensional image atoms instead of 2-D pixel grids. These atoms could serve a multitude of processing tasks, from super-resolution, depth estimation, to potentially light field editing. The proposed method is introduced in the following two chapters.

3 Steered Mixture-of-Experts

3.1 Introduction

This chapter introduces the novel *Steered Mixture-of-Experts* (SMoE) framework. SMoE is a unifying framework for representing any-dimensional image data while keeping the representation model both informative and functional. In the previous chapter, the current challenges were identified in order to make light field technology practically feasible. The challenging aspect is mainly due to the enormous amounts of pixel data. In order to face these challenges, the SMoE representation model is designed to be compact while inherently possessing beneficial properties for streaming and consumption of camera-captured scenes, such as: (1) random access, (2) pixel-parallel light-weight decoding, and (3) intrinsic view interpolation and super-resolution due to the continuous representation.

The proposed approach does away with the idea of dense 2-D grids of pixels as the core atoms of visual data. In the proposal, the continuous underlying pixel-generating functions that could have given rise to these pixel grids are approximated. Coherent areas in the higher-dimensional plenoptic function are represented by single higher-dimensional entities, called *kernels*. These kernels hold spatially-localized information about light rays at any angle arriving at a certain region. The global model consists thus of a set of kernels which define a continuous approximation of the underlying plenoptic function, or lower-dimensional projections of that function. In several short publications, SMoE was presented for approximation and coding of various image modalities including 2-D images,

360-degree images, video, light fields, and light field videos [54]–[59]. A longer and more detailed journal article containing an introduction to SMoE and the application towards light field images and video was also recently published [60]. Other papers focused more on the parallel and real-time decoding of SMoE models [18], [61], and one publication employed SMoE even as a separate color model for classical image and video formats [62].

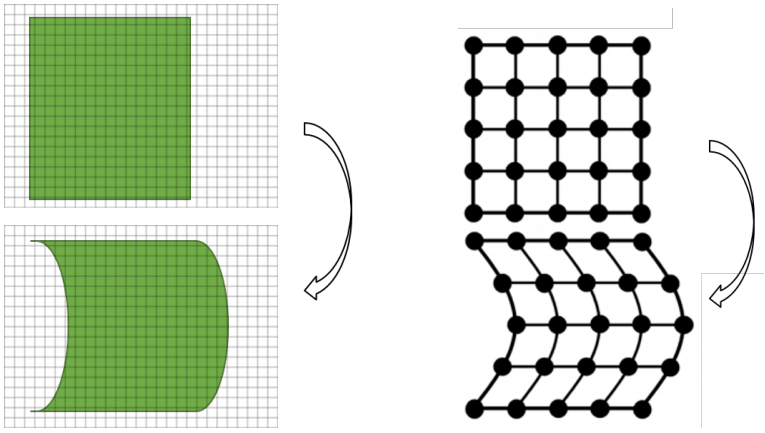
This chapter is structured as follows. First, a motivation is built based on the literature from fields in image processing and machine learning. Second, an overview of related models in the literature is provided and discussed. Third, the mathematical theory of SMoE is introduced. Fourth, the application of the SMoE model on 2-D images is illustrated and discussed in-depth. Finally, several experiments are presented to validate early design choices within the proposed framework.

3.2 Motivation

In general, two families of data representations exist: dense Eulerian and sparse Lagrangian. This distinction is common in e.g. the field of fluid dynamics [63]. Fig. 3.1 illustrates the difference between the two. Eulerian models are based on fixed observation points e.g. traditional raster images. The representation is fixed once and for all. In contrast, a Lagrangian approach attempts to construct a model of the intrinsic structure or topology in an image, i.e. the edge curves and/or the underlying object's discontinuities [64]. Often this requires that the intrinsic structure is to be estimated from the Eulerian representation produced by common capturing technology, e.g. samplers, sensors, ... Once the Lagrangian model is available, it is expected that the mid-level representation is much more powerful than the Eulerian one due to knowledge of the structure. Eulerian methods have the advantage that the location and the size of the sample amplitudes are known because of the grid-structure.

Current image processing and coding systems are mainly based on the Eulerian approach. Image coding is overwhelmingly dominated by strategies embedded in JPEG and JPEG-2000 using *discrete cosine transform* (DCT)- or wavelet-based transform-domain redundancy reduction techniques [11], [16], [65]. Pixels are aligned into dense sample grids and are processed or compressed as a single entity. These discrete sample grids frequently correspond to the camera architecture. For image coding, an inherent disadvantage of Eulerian approaches is that the spatial frequency bandwidth is uniform over the whole image. They also inherently suffer from the curse of dimensionality, e.g. the number of samples grows exponential with the number of dimensions.

Some research has gone into Lagrangian methods for coding images, e.g. segmentation-based coding for images and video [66], [67]. However, the over-



(a) Eulerian (b) Lagrangian

Figure 3.1: Illustration of the difference between Eulerian and Lagrangian representations (adapted from [63]). Eulerian data representations typically yields dense sampling grids of the data (or another fixed structure of observation locations). This is in contrast to Lagrangian representations in which an underlying structure is sought in terms of atoms of some sort. Changes recorded to the data over time are expressed as a new grid-like snapshot in Eulerian representations. Whereas in Lagrangian representations, the changes are modeled on the identified underlying entities.

head of signaling the segmentation borders was too high to be practical and block-based methods were preferred. The proposed SMoE approach is a Lagrangian representation which can similarly be interpreted as a segmentation-based coding technique. The proposed approach however, does not require explicit coding of the segmentation borders, as the segmentation is implicit as discussed in Sec. 3.4.2.

This dissertation is inspired by the works of Mumford-Shah, Prandoni & Vetterli, Takeda [69]–[71], and Lagrangian methods in general. First, the Mumford-Shah variational model assumes that natural images are characterized by having regions that behave smooth but are separated by discontinuities (edges) [69]. This behavior is described in a functional term for a loss function to be minimized in an optimization setting. This functional is frequently used for the segmentation of images. Fig. 3.2 illustrates a piecewise smooth approximation of a natural image based on the Mumford-Shah functional [68]. The image clearly shows the most informative parts of an image are well reconstructed as a piecewise smooth approximation, but the fine texture and details are lost by this simplification. From this observation, the assumption is that images have a piecewise stationary nature rather than a piecewise smooth. The term ”stationary” here refers to the property of a statistical process in which the process parameters do not change over time, i.e. it is if it were resulting from a stable single source. Nevertheless, a piecewise

(a) Input image (b) Piecewise smooth approximation

Figure 3.2: Example of a piecewise smooth approximation based on the Mumford-Shah functional [68]

smooth approximation already comes close to the original and is assumed to be a good starting point for a Lagrangian representation. Similar observations can be found when analyzing other image modalities. For example, in video, motion is approximated by line segments, e.g. motion vectors in video coding. Similarly, as shown in Chapter 2, light fields also exhibit linear structure along the EPI strips. From these observations, the underlying assumption is that pixels are instantiations of a non-linear or non-stationary random process that can be modeled by spatially piecewise-stationary processes. As such, the model takes into account different regions of the pixel data and their segmentation borders.

Secondly, Prandoni & Vetterli published theoretical and experimental work on the approximation and compression of piecewise smooth functions. In this work, they showed that for such functions, a sparse coding scheme is much more efficient than fixed grids [70]. Imagine the simple case of a piecewise smooth function, e.g. a step function: $y = \text{sign}(x)$. A dense representation would sample this function at regular intervals, having a long list of zeros followed by a list of ones. However, using a sparse representation one could just indicate a zero element, a one, and where the discontinuity lays on the x-axis. The secondary benefit of this is that the structure of the signal is known in the compressed domain, e.g. the location of the discontinuities.

Lastly, Takeda et al. introduced *Steered Kernel Regression* (SKR) for image processing in which kernels were steered along image features in order to perform edge-aware denoising, super-resolution, deblocking, and other image processing tasks [71]. Fig. 3.3 illustrates the steering along edges by the kernels. In this work, a sparse representation was pursued under the assumption that image modalities have a piecewise structure. As such, the work of Takeda's steering kernels gave us a first idea on how such a sparse representation could look like. Early work that lead to the development of SMoE used Takeda's framework for image coding [72].

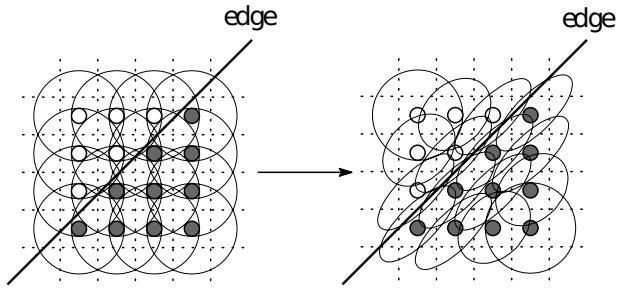


Figure 3.3: The "steering" of kernels along edges in Takeda's steered kernel regression framework [71].

The proposed approach was implemented as a pre- and post-processor for JPEG. The results were promising but it still relied on dense representations and did not scale to higher dimensions.

In the proposed method, these ideas are combined to represent the coherent regions in image modalities by a sparse set of kernels. The introduced SMoE approach also borrows many concepts from non-linear regression techniques in the machine learning world in which kernel approaches are well-known, i.e. *radial basis function networks* (RBF) [73] and non-linear *support vector regression* (SVR) [74]. The relation to such early machine learning models in the literature is discussed in the following section.

3.2.1 Related models in the literature

During the development of the SMoE framework, it came to my attention that other work had similar intention or used the same model but for other applications. In this section, all methods are grouped that were identified to have considerable overlap with this work in terms of model design. As the first references are over thirty years old, the list is by no means exhaustive but it is intended to provide a high-level overview of the evolution over time.

1988 Networks of locally receptive fields [75]: A neural network with a single layer of radial-symmetric locally-receptive neurons was proposed in contrast to sigmoidal activation functions. This bears similarity as it operates locally in the input space and enabled easier more tangible modeling not based on backpropagation.

1988 Radial basis functions, multi-variable functional interpolation and adaptive networks [73]: The work similarly proposes single-layer neural networks using locally-supported radial basis activation functions which coined the term RBF networks.

1989 Mixtures of linear regressors [76]: This work proposes learning a predecessor MoE as a single layer network that consists of only linear regressors. The EM-algorithm can be used in order to find these parameters [77].

1991 Adaptve mixtures of local experts [78]: The first proposal of the modular MoE concept which connects multiple subnetworks using gating functions. Each subnetwork focuses on a specific subtask. The idea is that for some tasks a set of simple networks is more adequate than a single big network.

1995 The alternative model for mixtures of experts [79]. This is the first model to exactly yield the same formulation of the MoE problem based on GMMs as is used in this dissertation. In this formulation, both the gates and the local expert functions are simultaneously derived from a GMM that models the joint probability density of the input and output space.

1995 Expectation-Maximization RBF (EMRBF) [80]: This work interprets RBFs as mixtures of univariate Gaussians which allows the application of known statistical tools including the EM algorithm for RBF parameter estimation. In contrast to earlier RBF networks which cluster only the input space, the input and output space is jointly clustered. This yields an identical formulation as the the alternative MoE model [79].

1996 RBF Network for non-linear image restoration [81]: This paper proposes a non-normalized Gaussian RBF network based on a GMM, while simultaneously being the first found usage of a similar model in the field of image approximation.

1998 Normalized Gaussian RBF Networks [82]: This work proposes the equivalent of the multivariate version of [80] in order to create a soft-segmentation of an any-dimensional input space.

2001 Image approximation and smoothing using SVR [83]: Similar image approximation was proposed that comes from the field of SVR. As such, the technique only retains the most important pixels (i.e. the support vectors) needed to form the image while tolerating a certain error margin. It is thus the first reference found that approximates the continuous function that maps the 2-D coordinate space of an image onto its gray-level amplitude.

2004 Gaussian Mixture Regression [84]: This work provides the derivation of a regression based on a GMM. The resulting formulas are identical to [79], however, it followed the same thought process as during the development of SMoE.

It is common that techniques are developed concurrently in different fields across the literature. It is not my intention to further scatter the literature, but

to provide an umbrella name for the distinct application of MoE models based on kernel entities that are steered along pixel correlation across any-dimensional image modalities. The framework is intended to be multipurpose - it does not only focus on approximation, or coding. The goal is to provide a model that is also descriptive as discussed in this work. This dissertation is mainly focused on the case of MoEs based on GMMs, but the methodology is not limited to GMMs and allows for future more expressive models.

As mentioned, the SMoE representation is further tightly linked to segmentation methods. Image segmentation in general is an active research field with many different approaches, specialized for diverse fields in computer vision. A recent overview on various approaches can be found in [85]. However, multidimensional segmentation remains challenging. The use of GMMs has been proposed for simultaneous multidimensional segmentation of image modalities, most predominantly, for spatio-temporal segmentation of video. Interestingly, GMMs were proposed for probabilistic space-time video modeling [86]. Similarly, the same model was used for video segmentation and compression using hierarchies of GMMs [87]. However, the GMMs were only used for the segmentation, not the coding aspect. The segmentation was consecutively used for guiding a more classical motion-compensated coding scheme. Similar to the proposed kernel representation idea is the work on identifying coherent 4-D atoms in light fields, i.e. super-rays [51]. The identification of super-rays then enables efficient light field processing as pixels over different viewpoints have a relation to each other. Moreover, dynamic super-rays have been proposed for light field video processing [88]. The SMoE approach bears a similar philosophy, but the SMoE approach abandons the concept of pixels altogether.

3.3 Steered Mixture-of-Experts

In SMoE, image modalities and in general signals are approximated by modeling them as a set of coherent kernels. Let us define a coordinate space $\mathbb{R}^p$ and a color space $\mathbb{R}^q$. For images, video, and 4-D light fields, the dimensionality p is respectively 2, 3, and 4. For monochrome images, q is 1, and for color images q is typically 3.

As mentioned in the previous section, the underlying assumption is that image pixels are instantiations of a non-linear or non-stationary random process that can be modeled by spatially-piecewise stationary stochastic processes. The goal is to divide the coordinate space X into stationary regions, and to find local regressors ($f : \mathbb{R}^p \mapsto \mathbb{R}^q$) that locally approximate this stationary region well. This is the general *Mixture-of-Experts* (MoE) strategy, well known in the machine learning field. However, SMoE is based on a mixture model (or "alternative") version of the MoE approach [89]. In this version, both segmentation and local regressors

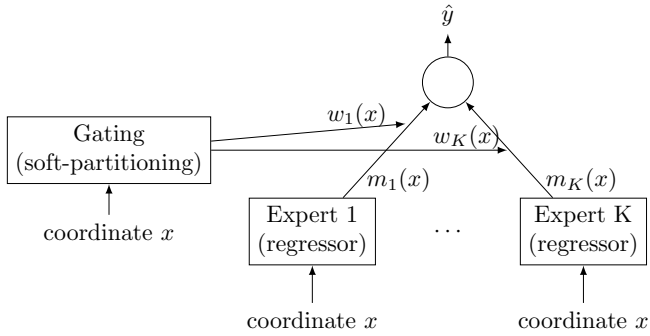


Figure 3.4: Illustration of a Mixture-of-Experts with one layer for regression. The gating function soft-partitions the input space in regions where particular experts (in this case regressors) are the most influential.

(the *experts*) are derived from the modes of a mixture model. This mixture model models the joint probability distribution of the random vectors $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$ in respectively the coordinate space and color space.

For now, let us focus on SMoE based on the *Gaussian Mixture Model* (GMM), other distributions are possible e.g. the *Student-t Mixture Model* (STM) (see Sec. 3.3.5). One Gaussian kernel in the model defines a linear regressor through the conditional $Y|X$, and all kernels combined define a segmentation of the coordinate space. The model thus only consists of a set of Gaussian kernels which are defined by their centers and covariances. The reason for choosing GMMs is that they offer elegant mathematics and limited parametrization. Furthermore, the MoE based on GMMs results in smoothed piecewise approximations, which is assumed to fit natural image modalities quite well, as mentioned above. However, the linear nature might fail to capture high spatial frequencies such as noise and fine texture. Therefore, we do not exclude future models with more expressive regressors.

The parameters of these kernels are found using likelihood optimization. Consequently, the kernels harvest correlation over all dimensions and steer along the dimensions with highest correlation. As such they align with e.g. edges (in spatial dimensions), temporal flow (in the time dimension) in the case of video, and EPI structures in light fields [54], [56], [60].

3.3.1 Mixture-of-Experts

In general, the goal of regression is to optimally predict a realization of a random vector $Y \in \mathbb{R}^q$, based on a known random vector $X \in \mathbb{R}^p$. Under the universal approximation theory, any reasonably well-behaving continuous function can be approximated by an artificial neural network [90]. MoEs are a type of committee machine, a neural network in which the responses of multiple neural networks are

combined into a single response [90]. More precisely, it falls under the category "committee machine with dynamic structure" as the weighing of the experts is dependent on the data.

The MoE tree structure is illustrated in Fig. 3.4. Given K experts with gate parameters $\Theta_g = \{\theta_{g,j}\}_{j=1}^K$, expert parameters $\Theta_e = \{\theta_{e,j}\}_{j=1}^K$, an input vector $\boldsymbol{x}$, and a target vector $\boldsymbol{y}$, the total probability of observing $\boldsymbol{y}$ can be written in terms of the experts, as

$$p_Y(\boldsymbol{y}|X=\boldsymbol{x},\Theta) = \sum_{j=1}^{K} \underbrace{p_X(j|\boldsymbol{x},\theta_{g,j})}_{\text{gate access}} \underbrace{p_Y(\boldsymbol{y}|\boldsymbol{x},\theta_{e,j})}_{\text{expert posterior}}. \tag{3.1}$$

Due to the modular structure, the gates can be placed in a tree-structure forming hierarchical MoEs (HME) [89]. The original MoE approach and modeling differentiated between the model parameters for the gates Θ_g and for the experts Θ_e, and relied on iteratively recursive least mean squares (IRLS) for estimating the expert parameters.

One reason of the popularity of MoEs is that these methods allow for *conditional computing*. Conditional computing is the process that allows for efficient calculations by only having to evaluate a limited number of branches of the tree [91]. As such, large portions of the tree are never evaluated and can significantly decrease the computational complexity. This principle is crucial for the the proposed SMoE model as it enables us to efficiently model extremely large datasets with billions of pixels (see Chapter 5), to reconstruct views in real-time (see Sec. 4.3.4), and to enable random access functionality [18].

SMoE is based on the "alternative" MoE definition which is deeply rooted in a Bayesian framework based on mixture models of distributions of the exponential family [89]. This method has the advantage that both the gates Θ_g and the experts Θ_e are simultaneously defined by the Gaussian components of the mixture model. Thus, the estimation of the parameters for gates and experts are optimized simultaneously and IRLS is not needed [89].

Consider a mixture of distributions. The joint probability is

$$p_{XY}(\boldsymbol{x},\boldsymbol{y}) = \sum_{j=1}^{K} \pi_j \phi_j(\theta_j), \tag{3.2}$$

with π_j being the prior for distribution ϕ_j.

Regressing the mixture model is equal to finding a measure of central tendency, such as the expectation or maximum-a-posteriori of Y given X of the mixture model, e.g. the mean, median and mode of the marginal $p_Y(Y|X=\boldsymbol{x})$. Note that the mean is the easiest to compute, and does not rely on the variance of $p_Y(Y|X=\boldsymbol{x})$. As such, less information needs to be transmitted in the case of coding. This

dissertation will focus on the expected value of the conditional $E[Y|X = \boldsymbol{x}]$, unless mentioned otherwise. In Sec. 3.5.3, the performance of the mean, median and mode estimators are evaluated for 2-D images.

3.3.2 Training of Mixture Models with Distributions of the Exponential Family

The *Expectation-Maximization* (EM) algorithm is frequently used for estimating parameters of a mixture model [92] in an unsupervised learning approach. In SMoE, the mixtures approximate the joint probability function $p_{XY}(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{p+q}$ of the input $\boldsymbol{x}$ and the output $\boldsymbol{y}$ vectors and implicitly define the parameters of gates Θ_g and the experts Θ_e of the MoE [93][94]. EM thus partitions the image coordinate and color space using a "divide and conquer" strategy and learns a linear regression surface on each partition.

The EM algorithm maximizes the loglikelihood, which in the case of the joint probability of X and Y given a mixture model of exponential distributions is given by

$$l(\Theta|X, Y) = E[\log p(\boldsymbol{x}, \boldsymbol{y}|\Theta)] \tag{3.3}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \log \sum_{j=1}^{K} \pi_j p_{XY}(\boldsymbol{x}_i, \boldsymbol{y}_i|\theta_j). \tag{3.4}$$

Assume an indicator variable z_{ij} which indicates the unknown true membership of $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ to the distribution j. In the so-called E-Step for mixtures of distributions of the exponential family, z_{ij} is estimated iteratively by the normalized exponential at iteration k as follows,

$$\text{(E-step) } \hat{z}_{ij} = \frac{\pi_j \phi_j(\boldsymbol{x}_i, \boldsymbol{y}_i; \theta_j^k)}{\sum_{i=1}^{K} \pi_i \phi_j(\boldsymbol{x}_i, \boldsymbol{y}_i; \theta_j^k)}. \tag{3.5}$$

Given the estimated soft-memberships, parameters Θ^{k+1} that approximate z_{ij} closest are derived from the M-Step:

$$\text{(M-step) } \Theta^{k+1} = \arg\max_{\Theta} \hat{z}_{ij}; \tag{3.6}$$

or more explicitly:

$$\pi_j = \frac{1}{N}\sum_{i=1}^{N} \hat{z}_{ij} \tag{3.7}$$

$$\boldsymbol{\mu}_j = \frac{1}{\pi_j}\sum_{i=1}^{N} \hat{z}_{ij} \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix} \tag{3.8}$$

$$R_j = \frac{1}{\pi_j}\sum_{i=1}^{N} \hat{z}_{ij} \left(\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix} - \boldsymbol{\mu}_j \right) \left(\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix} - \boldsymbol{\mu}_j \right)^T \tag{3.9}$$

The optimization problem is unfortunately non-convex and converges to a local optimum [93]. Consequently, EM is sensitive towards the initialization of the parameters of the experts, i.e. the positions and steering. In SMoE, possibly billions of pixels are fitted by hundred thousands of kernels. The implementation of the EM algorithm on such a scale thus requires tremendous care and optimizations. Chapter 5 is dedicated to the modeling of extremely large mixture models.

3.3.3 Mixture-of-Experts based on GMMs

GMMs offer elegant and relatively-easy descriptions for distributions and are frequently used to approximate multi-modal, multi-variate distributions $p_{XY}(\boldsymbol{x}, \boldsymbol{y})$. Given a GMM, one can derive a regression as follows [82][84]. Assume data $D = \{\boldsymbol{x}^i, \boldsymbol{y}^i\}_{i=1}^N$ with joint probability density:

$$p_{XY}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\mu}_j, R_j) \tag{3.10}$$

and $\sum_{j=1}^{K} \pi_j = 1, \boldsymbol{\mu}_j = \left[\begin{smallmatrix} \boldsymbol{\mu}_{X,j} \\ \boldsymbol{\mu}_{Y,j} \end{smallmatrix} \right], R_j = \left[\begin{smallmatrix} R_{XX,j} & R_{XY,j} \\ R_{YX,j} & R_{YY,j} \end{smallmatrix} \right]$.

The parameters of this model are $\Theta = [\theta_1, \cdots, \theta_K]$, with $\theta_j = (\pi_j, \boldsymbol{\mu}_j, R_j)$, respectively being the priors, centers, and covariances. A normal *probability density function* (pdf) of dimension $p + q$ can be factorized as

$$\mathcal{N}_{p+q}\left(\begin{bmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_Y \end{bmatrix}, \sigma^2 \right) = \mathcal{N}_q(\boldsymbol{\mu}_{Y|X}, R_{Y|X}) \mathcal{N}_p(\boldsymbol{\mu}_X, R_{XX}),$$

where $R_{Y|X}$ is the *Schur* complement:

$$R_{Y|X} = R_{YY} - R_{YX} R_{XX}^{-1} R_{XY}. \tag{3.11}$$

Accordingly, the factorization for a mixture becomes:

$$p_{XY}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{K} \Big[\pi_j \mathcal{N}_{Y|X,j}(\boldsymbol{y}; m_j(\boldsymbol{x}), R_{Y|X,j}) \tag{3.12}$$

$$\times \mathcal{N}_{X,j}(\boldsymbol{x}; \boldsymbol{\mu}_{X,j}, R_{XX,j}) \Big], \tag{3.13}$$

with

$$m_j(\boldsymbol{x}) = \boldsymbol{\mu}_{Y,j} + R_{YX,j}R_{XX,j}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{X,j}). \tag{3.14}$$

$m_j(\boldsymbol{x})$ defines one of the above mentioned $\mathbb{R}^p \mapsto \mathbb{R}^q$ expert functions, which in the GMM case are q linear functions. The slope is defined by $R_{YX,j}R_{XX,j}^{-1}$. If steered, i.e. non-homogeneous Gaussians in GMM are used, the desired linear steering kernels for SMoE are obtained. Each kernel adapts to local statistics but - in contrast to RBFs, SVR and SKR - each kernel has global support over the entire signal domain.

The MoE approximation function is derived from the conditional pdf $Y|X$ [84]

$$p_Y(Y|X = \boldsymbol{x}) = \sum_{j=1}^{K} w_j(\boldsymbol{x})\mathcal{N}(\boldsymbol{x}; m_j(\boldsymbol{x}), R_{Y|X,j}), \tag{3.15}$$

with mixing weights

$$w_j(\boldsymbol{x}) = \frac{\pi_j\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{X,j}, R_{XX,j})}{\sum_{i=1}^{K}\pi_i\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{X,i}, R_{XX,i})}. \tag{3.16}$$

Note that the MoE gating function in Eq. 3.16 corresponds to the normalized exponential or the *softmax* function frequently used in ANNs. It defines the support region for each kernel and ensures that each sample has support.

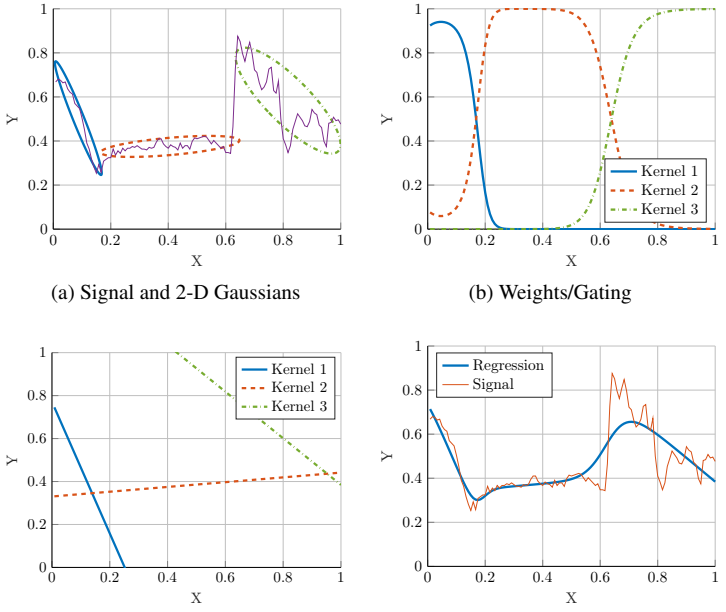
Non-linear regression is enabled by calculating the expected value $\hat{\boldsymbol{y}}$ given a sample location $\boldsymbol{x}$ through the conditional. From Eq. 3.15 and 3.16 follows the *non-linear regression function* $m(\boldsymbol{x})$:

$$\hat{\boldsymbol{y}} = m(\boldsymbol{x}) = \mathbf{E}[Y|X = \boldsymbol{x}] = \sum_{j=1}^{K} w_j(\boldsymbol{x})m_j(\boldsymbol{x}). \tag{3.17}$$

The trustworthiness of the prediction of the ith component in Y, can then be evaluated by calculating the prediction variance $\mathrm{var}[Y^i|X = \boldsymbol{x}]$.

3.3.4 Example: 1-D Steered Mixture-of-Experts (SMoE)

For illustration purposes, Fig. 3.5 depicts a SMoE regression of samples from a 1-D image scan line. The Gaussians/kernels were optimized using the EM algorithm. Notice that both X and Y are 1-D, we thus estimate 2-D pdfs using steered Gaussians. In Fig. 3.5a, the Gaussians in the GMM are visualized as ellipses, which indicate iso-probability. Each Gaussian is responsible for a region in X defined by the weights (Eq. 3.16), as visualized in Fig. 3.5b. Fig. 3.5c shows that each kernel also yields a linear regressor based on the expectation of the conditional $Y|X = \boldsymbol{x}$. Finally, the resulting *smoothed piecewise linear* regression function by the weighted sum over all kernels is shown in Fig. 3.5d.



(a) Signal and 2-D Gaussians

(b) Weights/Gating

(c) Linear regressors

(d) Regression vs. signal

Figure 3.5: A 1-D regression example using SMoE on a part of a scanline taken from Lena *using three Gaussian kernels ($K = 3$). The 2-D GMM models the joint probability between X and Y (a). From this model, a gating function (b) and the regressors (c) are derived. The regressors are summed according to the gating function to yield the regressed function in (d).*

3.3.5 Student-t Mixture Models

Gaussian distributions are beloved for their mathematical simplicity and practicality. However, Gaussian distributions are known to be sensitive towards outliers. Research has provided evidence for the hypothesis that STMs may be better suited than GMMs for modeling natural images [95]. The regression of an STM can be derived analogous to the regression based on GMMs. Student-t distributions have the same conditional expectation $E[Y|X]$ as the Gaussian distribution, as such the gradient m_j for every component j is identical to the GMM case [96]. The regression does have different mixing weights:

$$w_j(\boldsymbol{x}) = \frac{\pi_j \mathcal{T}_j(\boldsymbol{x}; \boldsymbol{\mu}_{X,j}, \Sigma_{XX,j}, \nu_j)}{\sum_{i=1}^{K} \pi_i \mathcal{T}_i(\boldsymbol{x}; \boldsymbol{\mu}_{X,i}, \Sigma_{XX,i}, \nu_i)}. \tag{3.18}$$

The t-distribution $\mathcal{T}$ is described by its degrees of freedom ν, a mean $\boldsymbol{\mu}$ (of

dimension $d = p + q$), and a symmetric matrix parameter Σ ($d \times d$) [96].

$$\mathcal{T}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma, \nu) = \frac{\Gamma(\frac{\nu+d}{2})}{(\nu\pi)^{d/2}\Gamma(\frac{\nu}{2})} \times \left(1 + \frac{1}{\nu}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right)^{-\frac{d+\nu}{2}},$$

with $\Gamma(\cdot)$ being the Gamma function.

The EM algorithm describes a general iterative structure for obtaining mixture models. For GMMs all calculations are closed-form. However, for STMs the parameter ν_j can not be calculated in closed form [95]. The parameter ν_j is the root of a non-linear equation. This can be found through e.g. Brent's method [97]. This comes at the expense of a constant factor in computational overhead compared to GMMs.

For illustration purposes, Fig. 3.6 depicts two SMoE models for regression of samples from a 1-D image scan line. The MoEs were optimized using the EM algorithm based on a GMM and an STM respectively. In the example the three distinct stationary regions were corrupted with Gaussian and speckle noise. It is apparent that STM allows for a more robust data clustering because of the longer tails. However, the resulting regression does not change significantly. In this example, GMM allocates one component to model the noise, whereas the STM covers the noise data points through the thicker tails of some of the components. Consequently, a strong overlap of components results in both cases: in GMM w_3 has effective global support, whereas in the STM example w_2 has effective global support. In Sec. 3.5.2, experiments are provided that compare GMMs and STMs for the approximation of 2-D images using SMoE.

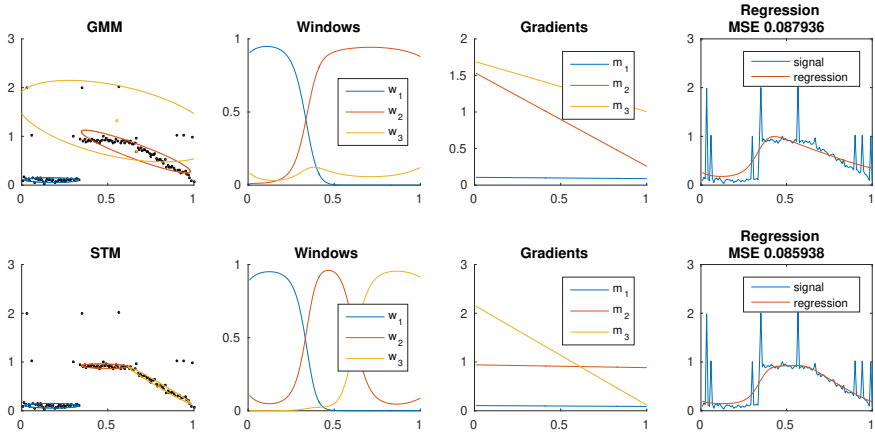


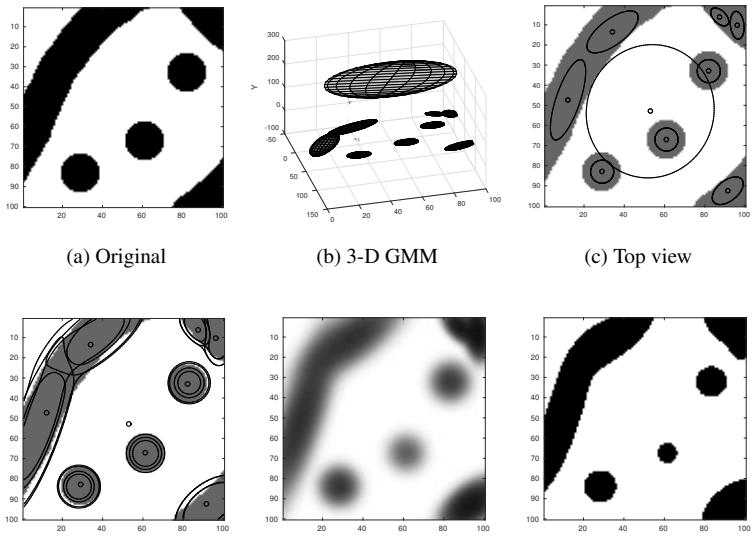
Figure 3.6: 1-D example of how STM and GMM handle outliers different on a signal corrupted with Gaussian and speckle noise.

3.4 Insights into the SMoE Image Model

In this section, the applications of the SMoE model approach are outlined for 2-D images for which a number of illustrative results are provided. Consequently, this provides easier understanding of the SMoE framework before introducing the extension to higher dimensional image modalities, e.g. 4-D light field images and video in Chapter 4.

3.4.1 SMoE for sparse image representation

For grayscale images, let us define $\boldsymbol{x}_i \in \mathbb{R}^2$ as the pixel coordinates (row, column) and $\boldsymbol{y}_i \in \mathbb{R}^1$ as the amplitudes of image pixels. Regressing the model is equal to finding the expected amplitude $\hat{\boldsymbol{y}}_i$ given a location $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}]$ through the "learned" conditional pdf, i.e. $\hat{\boldsymbol{y}} = m(\boldsymbol{x})$. Each kernel defines a linear expert function $\mathbb{R}^2 \mapsto \mathbb{R} : m_j$ as their regressor, which visually describes a gradient per kernel. The gradient indicates how the signal behaves around the center of the kernel (Eq. 3.14). Furthermore, each kernel defines a 2-D window gating function $\mathbb{R}^2 \mapsto \mathbb{R} : w_j$, which defines the operating region, or support of the expert. The window function w_j gives weight to each sample, indicating the soft membership of that pixel to that component (Eq. 3.16). Note that by jointly modeling the pixel locations and amplitudes, the kernel windows can steer along edges and adapt to regional signal intensity flow, similar to the locally-supported SKR [71].



(a) Original (b) 3-D GMM (c) Top view

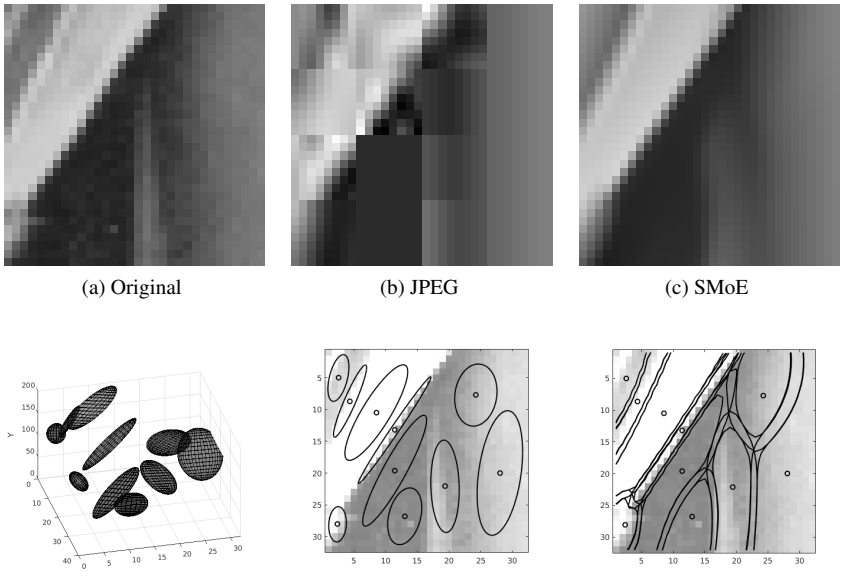
(d) Gating/Windows (e) Mean estimator (f) Hard threshold

Figure 3.7: Example of a black-white image (a) modeled by 9 kernels for 10.000 pixels (1 kernel covers ± 1111 pixels on average). The kernels are visualized in (b) in $\mathbb{R}^3$ joint coordinate and color space. In (c) the spatial spread of the kernels is shown as an overlay on the original image. Illustration (d) shows the mixing weight w_j (or responsibility) of each kernel j on each pixel after softmax. The continuous regression is shown in (e) and the regression quantized into 1 bit in (f). Note how the kernels in (b) are virtually flat in the Y dimension as they represent constant colors.

3.4.1.1 Example: binary image

Let us use the binary image in Fig. 3.7a to illustrate the approximation of the binary pixel values using only a very small number of experts, $K = 9$ kernels. The GMM (after learning using EM) results in the 3-D mixture model illustrated in Fig. 3.7b and Fig. 3.7c. Due to the fact that only two luma values are present in the image, the estimated 3-D ellipsoids are flat along the Y-dimension, i.e. $R_{YY,j}$ and $R_{YX,j}$ are zero for each component j. From Eq. 3.14 and Eq. 3.11, it results that the regressors defined by each component are constant planes based on non-homogeneous kernels, and the conditional variance is zero. Remarkably, the full background is covered by a single large white kernel. While all other experts are gated to provide only local support within this image, one expert provides local as well as global support in X.

The gating windows are shown in Fig. 3.7d and confirm the directional steering operation of the experts. The windows softly overlap while forming arbitrarily-



(a) Original (b) JPEG (c) SMoE

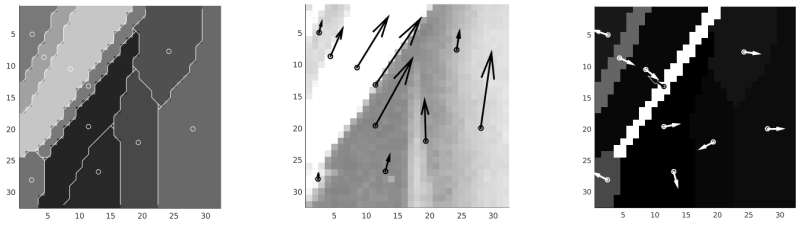
(d) 3-D GMM (e) Top view (f) Gating/Windows

Figure 3.8: Example of SMoE modeling and reconstruction on a 32x32 pixel crop from Lena *(a). The kernels are visualized in joint coordinate and color space $\mathbb{R}^3$ (d) and as an overlay to the image only showing the spatial spread in X (e). (f) illustrates how these kernels are responsible for irregularly-shaped regions after softmax. At the same bitrate, JPEG (b) results in block artifacts whereas SMoE has a reconstruction (c) that is smoothed along image features. Note how each component covers a range of luma values in Y and the corresponding regressors thus result in gradients.*

shaped segments. When the expected value of the conditional of the mixture model is calculated (*mean estimator*), we arrive at a continuous-tone image shown in Fig. 3.7e. A binary image can be obtained in two ways: (1) by hard thresholding as illustrated in Fig. 3.7f, i.e. mapping all luma values $y \geq 0.5$ to be white, and all luma < 0.5 to be black, or (2) by using the mode of the conditional pdf (*mode estimator*). Even though only $K = 9$ kernels are used to represent the image content, it is clear that all "objects" are represented. It is apparent that image approximation using SMoE results in geometrical distortion of image objects.

3.4.1.2 Example: natural image

Fig. 3.8 illustrates the modeling and reconstruction of a 32x32 pixel crop of *Lena* using $K = 10$ components. The SMoE model parameters were quantized prior to reconstruction to arrive at a designated bit rate in order to allow a comparison with JPEG (Fig. 3.8b) as a simple compressed and coded representation. For fair



(a) Segmentation (b) Intensity flow (c) Edges

Figure 3.9: The decoded coefficients provide MPEG-7-like descriptors.

comparison, the bits required for the JPEG header were subtracted. Both representations are at around 0.35 bit/sample [54].

Comparing Fig. 3.8b and Fig. 3.8c, it is apparent that especially the edges are reconstructed with impressive quality and sharpness by the SMoE approach. It can be argued that SMoE provides for a much more efficient and sparse image representation than JPEG for this type of image content. Fig. 3.8d shows the steering of the 3-D ellipsoid Gaussian "cigar" kernels, which define the m_j global 2-D gradient planes for regression. Fig. 3.8e illustrates the ellipsoids projected onto the 2-D pixel domain. As intended, the SMoE kernels harvest directional pixel correlation as the kernels capture spatial structure.

The respective window functions dictate how the kernel gradients are gated. The windows overlap adaptively into adjacent image regions and enable either smooth transitions between regions or abrupt changes. The windows are of arbitrary shape and steer along edges. This assures that dominant edges are well reconstructed considering the low amount of kernels. The sparse representation thus prioritizes dominant structures over smaller details, as is the goal of any sparse image representation. Note that the dominant gradient on the right is very well approximated by a single kernel. Fine details and noise are eliminated which is the result of the very sparse representation with only 10 kernels.

3.4.2 Image descriptors

One of the advantages of using a sparse Lagrangian representation is that the model itself exhibits the structure of the data. Consequentially, the SMoE model includes novel MPEG-7-like image descriptors solely based on the kernel parameters [15]. When images are compressed/stored in SMoE format this information is readily available for several (decoder) post-processing tasks. This may include tasks such as segmentation, noise reduction, scale conversion, image similarity retrieval, classification and object recognition to name a few.

Figure 3.10: An example of mean estimated reconstructions of a 128x128 image from the dataset. Original (left) followed by models with 25, 100, 250, 750, and 2000 components, i.e. ranging from 1 kernel covering ± 655 to ± 8 pixels on average.

Fig. 3.9 provides three illustrations of descriptors of the SMoE model shown in Fig. 3.8 in the case of 2-D images: segmentation, intensity flow, and edge detection. First, since the SMoE approach follows a Bayesian interpretation, a segmentation of the image into K regions can be easily obtained by deriving the maximum posteriori probability of each image pixel from the window functions w_j. The segmentation boils down to determining for each pixel the most dominant kernel. Secondly, the intensity flow can be derived from the kernel parameters. The intensity flow here can be seen as the local orientation of a component and is thus the principle component of the decoded coefficients in $R_{XX,j}$. Finally, the SMoE model contains valuable p-dimensional gradient information and allows us to perform multi-dimensional edge detection based on the kernel parameters. Let us define edge strength as the slope strength $|S_j|$, with S_j being the slope $R_{YX,j}R_{XX,j}^{-1}$. Furthermore, even the orientation of the edge, i.e. the orientation of the local gradient is given by the decoded principle component of S_j using an Eigen-decomposition. Elaborate discussion on the extended additional functionality that SMoEs may enable is beyond the scope of this dissertation. However, it is clear that the model is extremely informative in the compressed domain, i.e. based on model parameters.

3.4.3 Color representation

When extended to support and regress color values in images, the output Y becomes a 3-D random variable (e.g. in case of the YCbCr color space). In this case the steered kernels are based on a "learned" 5-dimensional pdf (2-D location, 3 color channels). During modeling, the 5-D kernels now explore correlation in horizontal and vertical dimensions as well as in 3-D color space. However, the regression for each channel is independent to each other. In consequence, each color channel has the same 2-D window w_j (Eq. 3.16), but different and independent regressors $m_{Y,j}$ (luma), $m_{Cb,j}$, and $m_{Cr,j}$ (chroma). In other words, each kernel describes a gradient in each color dimension:

$$m_{\mathrm{Y},j}(\boldsymbol{x}) = \boldsymbol{\mu}_{\mathrm{Y},j} + R_{\mathrm{Y}X,j}R^{-1}_{XX,j}(\boldsymbol{x} - \boldsymbol{\mu}_{X,j}), \tag{3.19}$$

$$m_{\mathrm{Cb},j}(\boldsymbol{x}) = \boldsymbol{\mu}_{\mathrm{Cb},j} + R_{\mathrm{Cb}X,j}R^{-1}_{XX,j}(\boldsymbol{x} - \boldsymbol{\mu}_{X,j}), \tag{3.20}$$

$$m_{\mathrm{Cr},j}(\boldsymbol{x}) = \boldsymbol{\mu}_{\mathrm{Cr},j} + R_{\mathrm{Cr}X,j}R^{-1}_{XX,j}(\boldsymbol{x} - \boldsymbol{\mu}_{X,j}). \tag{3.21}$$

Fig. 3.10 illustrates the extension towards color images. the capability of SMoE for approximating images with varying levels of sparsity is demonstrated (number of kernels K between 25 and 2000). Fig. 3.10 thus illustrates that SMoE provides a continuously-refined low-pass version of the original.

Alternatively, the correlation between luma and chroma can be used in order to calculate chroma from luma. It is known that luma and chroma are locally linearly correlated [62]. The idea is that it requires more coefficients to store the covariance between the coordinates and the color channel compared to the covariance between the color channel and luma, i.e. p coefficients per color channel vs. one scalar value per color channel and one scalar indicating the variance of the luma component. As such, the color gradients are dependent on the estimated luma value, the covariance between luma and chroma, and the variance of the luma component in Y as follows:

$$m_{\mathrm{Cb},j}(y) = \boldsymbol{\mu}_{\mathrm{Cb},j} + \frac{\sigma_{\mathrm{YCb},j}}{\sigma^2_{\mathrm{YY}}}(y - \boldsymbol{\mu}_{\mathrm{Y}}), \tag{3.22}$$

$$m_{\mathrm{Cr},j}(y) = \boldsymbol{\mu}_{\mathrm{Cr},j} + \frac{\sigma_{\mathrm{YCr},j}}{\sigma^2_{\mathrm{YY}}}(y - \boldsymbol{\mu}_{\mathrm{Y}}). \tag{3.23}$$

Note that the input of these functions are thus 1-D, instead of 2-D compared to independently regressing each color channel. In the case for 2-D images, with three color channels, the gain is limited. In the first case of three independent regression, three 2-D covariances (6 coefficients) are thus required. The second case of chroma-from-luma prediction thus requires one 2-D covariance for the luma prediction, two scalar covariances $\{\sigma_{\mathrm{YCb}}, \sigma_{\mathrm{YCr}}\}$ and the luma variance σ^2_{YY} (5 coefficients). As such, using this prediction requires one coefficient less. However, the gain increases in the case of more color channels and in higher-dimensional imagery as p increases, while the number of color channels does not increase. The main disadvantage is that there is error accumulation as the chroma is predicted from a predicted luma. Experiments evaluating both methods and the case of using constant chroma regressors is provided in Sec. 3.5.4.

Interestingly, SMoE can thus be used as a separate color model [62]. Given a pixel coordinate $\boldsymbol{x}$ and its corresponding luma value Y, we can predict its chroma components. As such, we can encode the luma channels for images using conventional mechanisms, e.g. JPEG and use a model with a low amount of kernels as a color model. As the softmax (Eq. 3.16) takes into account the luma value as an

extra coordinate dimension, kernels are weighted simultaneously by their spatial distance as well as the difference in luma. It was shown that the model is reusable over a considerable number of frames while tolerating limited motion [62].

3.4.4 Resampling, pixel-parallel reconstruction and random access

In this section, the continuous nature of the model and its benefits are highlighted, i.e. inherent resampling, pixel-parallel reconstruction and random access. Firstly, the SMoE representation is fit onto the discrete pixel data during modeling. Nevertheless, the resulting regression is continuous. This is useful for resampling e.g. super-resolution of the spatial dimension (upsampling) or changing frame-rates in the case of video. In SMoE, resampling is merely evaluating the regression functions at different coordinates. No extensive investigation of the resampling capabilities of SMoE for 2-D images has been performed yet. However, Chapter 4 illustrates resampling along the camera dimensions enables view interpolation and extrapolation in the case of 4-D light fields.

Secondly, a major benefit is that each pixel can be decoded independently from other pixels. Once all kernel parameters are decoded, each pixel is only dependent on the kernel parameters. This allows for extremely practical pixel-parallel real-time decoding on GPUs and is extensively evaluated in the works by Avramelos et al. and Saenen et al. [18], [61].

Finally, the tree-like structure of MoEs enables for conditional computing by only evaluating a limited number of branches of the MoE tree [91]. In our case, there is only a single level, and a single branch corresponds to a single expert function. As such, the relevant branches, i.e. kernels to be evaluated can be identified cheaply as the gating function takes only into account the pixel coordinate and many branches do not need to be evaluated. In other words, as our gating operates directly on image coordinates, it is easy to devise quick heuristics on which kernels are likely to be relevant and which ones are not. In practice, a single pixel is only dependent on kernels in its vicinity. As such, this allows for random access functionality. If only a subset of pixels is required, then only a subset of kernels needs to be decoded. Furthermore, the conditional computing also enables scalability of the reconstruction speed. If hard real-time constraints are requested, then the number of kernels to be evaluated can be limited in order to ensure a fixed framerate [61].

3.5 Image Experiments

Let us evaluate the reconstruction capability of SMoE in terms of *peak signal-noise ratio* (PSNR) and *structural similarity* (SSIM) in this section [98]. The quality

of images is examined at different sparsity levels (various K), different density models (GMM vs STM), and by using different reconstruction methods (mean, median, mode). These experiments have not been published before.

3.5.1 Dataset

The dataset consists of 484 images of size 128x128. All images are crops from the Kodak Lossless True Color Image Suite (360 crops), and 124 crops from the standard color image test set consisting of *Lena*, *Baboon*, *Peppers* [99]. The Kodak Image Suite contains images with digitally added borders. To assure that the set contains purely natural images, borders of 64 pixels at the edges were discarded. PSNR and SSIM are used as metrics for evaluating the image quality. SSIM is calculated on the Y-plane only.

3.5.2 GMM vs STM

In this experiment, GMM modeling and regression is compared with a mixture of t-distributions. To this end, the 128x128 images are modeled using the same initialization. Regression of the images is performed based on Eq. 3.17, and the difference in quality is evaluated between the reconstruction of GMM and STM models in terms of ΔPSNR and ΔSSIM. For each configuration i, i.e. (image, K)-tuple, the resulting $\Delta\,\mathrm{PSNR}^i$ is $\mathrm{PSNR}^i_{STM} - \mathrm{PSNR}^i_{GMM}$ is calculated. The models are of size $K = [25, 100, 250, 750]$.

STMs have been shown to be better at modeling natural image statistics than GMMs [95]. This is confirmed by looking at the mean loglikelihood for each number of components in Fig. 3.11. Loglikelihood, PSNR and SSIM are generally correlated within SMoE, i.e. a better fit of the model results in better reconstruction quality. However, the experiments show that a better model fit (in terms of loglikelihood) does not necessarily result in better image quality. This is depicted by the mean PSNR and SSIM differences between GMM and STM regressions in Fig. 3.11. However, differences are relatively small.

In terms of computational complexity, the experiments showed that STM needed twice as long to compute the same amount of iterations than GMMs of the same size. In addition, the number of parameters grows as every component holds an extra parameter, i.e., the degrees of freedom ν_j. It can be concluded that from a practical viewpoint GMMs are better suited for mixture regression for 2-D images.

3.5.3 Mean/median/mode estimators

In this section, the reconstruction quality is evaluated in terms of PSNR and SSIM for different reconstruction strategies. In Sec. 3.3.1, it is described how the joint

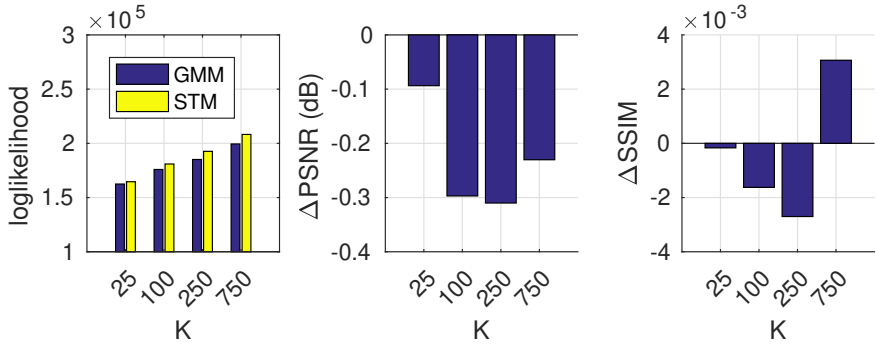


Figure 3.11: Comparison of STM and GMM with K kernels on the dataset. The difference is reported given their mean loglikelihood (left) and the mean quality gain of STM compared to GMM in terms of PSNR (middle) and SSIM (right). It is clear that STM provides a better model fit in terms of loglikelihood, although this does not translate consistently to a better reconstruction. In general, the effect is small, especially for SSIM.

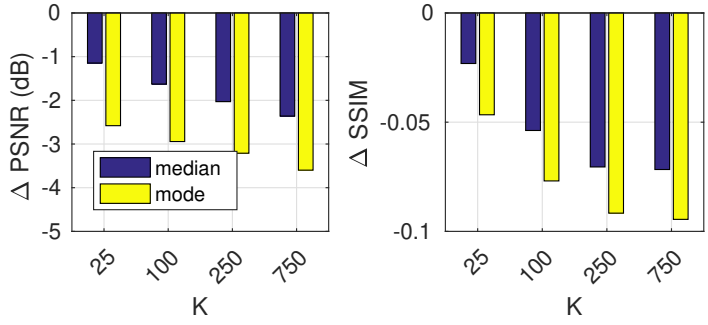


Figure 3.12: The mean difference of objective quality (ΔPSNR, ΔSSIM) of the median and the mode estimator relative to the mean estimator.

likelihood model gives rise to various regressions depending on what is chosen for the expert functions. In general, any measure of central tendency can be chosen. In this experiment, the mean, the median and the mode estimators are compared for the same dataset as above with $K = [25, 100, 250, 750]$ kernels. Fig. 3.12 depicts the difference in terms of PSNR and SSIM compared to the mean estimator. The results show that median and mode estimators generally result in a considerable loss of objective quality in terms of PSNR, and relatively low loss in terms of SSIM.

Fig. 3.13 illustrates the visual results of the three different estimators for the used dataset. The choice of reconstruction technique depends on individual preference and/or application. The median and mode estimators result in sharper images with a more artificial appearance which may be used for CGI, cartoons, or other

Figure 3.13: Examples of reconstructions on models of various size K (top to bottom: K=25,250,750,750,250,750). From left to right: original, mean, median, and mode.

artistic purposes. However, unless otherwise stated, this dissertation is centered around the mean estimator for two reasons. Firstly, from above experiments it leads to higher objective quality. Secondly, the mean is computationally cheaper and does not rely on the prediction variance. The mean is therefore not dependent on R_{YY}, and thus R_{YY} does not need to be transmitted.

3.5.4 Chroma reconstruction

As mentioned in Sec. 3.4.3, the chroma planes can be estimated as two additional regressions (Eq. 3.20 and Eq. 3.21) or chroma can be predicted from the luma regression (Eq. 3.22 and Eq. 3.23). The latter requires less data storage as $R_{\mathrm{Cb}X}$ and $R_{\mathrm{Cr}X}$ are vectors with p components, while σ_{CbY} and σ_{CrY} are scalars. However, it does introduce error propagation as the chroma is predicted from a luma estimate. In this experiment, three methods for regressing the color from the same GMMs are evaluated. The first method performs three independent regressions and requires most coefficients from the model. The second method is luma-to-chroma prediction, which requires less coefficients. The last method is using constant chroma functions per kernel. Consequently, the chroma regression is smoothed piecewise constant in the third case. This method relies on the least coefficients as it does not require any chroma covariance coefficients. As such, it only allows a gradient in the luma channel.

Fig. 3.15 shows the mean difference between reconstructing each color channel independently with chroma-from-luma prediction, as well as the comparison with the case that the chroma values are assumed to be constant. It is clear that both approaches have relatively minor impact on the total reconstruction, with a maximum mean loss of -0.3 dB $\mathrm{PSNR}_{\mathrm{Cr}}$. Chroma prediction performs generally better than constant chroma, but it can also do more harm than good as it suffers from error propagation. Fig. 3.14 illustrates a color artifact caused by luma-to-chroma prediction. Furthermore, careful implementation is needed. Artifacts can arise when σ_{Y}^2 becomes too small. Consequently the estimated gradient becomes unstable. This can happen when a component consists of a flat surface, e.g. in the case of overexposed areas. In Fig. 3.14, it was solved by adding 0.001 to the denominator. Note that this only makes the calculation numerically stable, but it does not eliminate the overly yellow parts.

It can be concluded that independent regression consistently yields the highest objective quality, although the loss of both alternatives remains relatively small. In general, errors in chroma are more tolerated than errors in luma values. Therefore, using the constant chroma regressors is an interesting option as it requires the least number of parameters, which allows for more efficient coding. Moreover, it does not require extra parameters when the dimensionality p increases.

Figure 3.14: Artifact when using luma-to-chroma prediction due to error propagation. Left: independent color plane estimation, right: luma-to-chroma prediction.

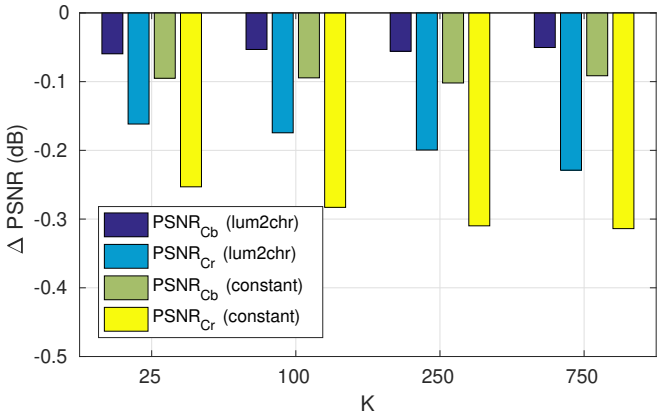


Figure 3.15: Mean loss in $PSNR_U$ and $PSNR_V$ using chroma prediction (lum2chr) and constant chroma for each component (constant).

3.6 Conclusion

In this chapter, the novel unifying SMoE framework for representing image modalities was presented. It was shown how the representation consists of a collection of kernels that represent coherent regions in the coordinate space of the image modality. The kernels represent multivariate distributions that could have generated the pixel values in that region. The concepts of SMoE were illustrated on 2-D images to provide insights into the geometrical interpretations of these kernels. Note, that one of the interesting concepts of the proposed approach is that as SMoE operates in the spatial domain and thus all parameters have geometrical interpretation.

The assumption is that image modalities consist of coherent regions that behave stationary and therefore a spatially-piecewise representation was pursued. The core concept of the proposed representation is that coherent regions in the coordinate space are represented by a single multidimensional Lagrangian entity, i.e. a kernel. The representation thus consists of a collection of kernels. Each multivariate kernel is responsible for a region in the coordinate space by evaluating the

likelihood of that kernel in the coordinate space. The regression is a weighted sum based on the conditional probability function of each kernel. In other words, each kernel is queried for that kernel's regression of each color channel for a certain pixel position. These functions are then weighted by the likelihood of that kernel being responsible for that pixel. The regression results in a smoothed piecewise reconstruction.

It was shown that the SMoE representation is a sparse continuous representation with several interesting properties. First, the kernels harvest correlation over every dimension and therefore steer in the direction of highest correlation. The representation thus takes on the structure of data itself. The representation becomes information-rich as it contains a mid-level understanding of the model, yielding several image descriptors, e.g. edge detection, segmentation, and intensity flow information. Secondly, in practice, pixels are independently reconstructed by kernels in their immediate vicinity, which enables pixel-level parallel reconstruction and yields potential for random access. Thirdly, resampling an image is merely resampling the representation without the need for any interpolation technique. Consequently, it can be concluded that SMoE provides an information-rich, feature-rich and compact representation. However, the model based on GMMs currently fails to capture high spatial frequencies, e.g. fine textures and noise. Therefore, the usage of more expressive representations should be explored in the future.

Additionally, a number of experiments were performed in order to support some design choices. As one of the main applications is coding, there is a clear benefit of representations that require few coefficients per kernel. There is thus a trade-off between reconstruction quality, computational complexity and number of parameters-per-kernel. Keeping these three requirements in mind, experiments have shown evidence for making some design decisions. First, it was concluded that GMMs are preferred over Mixture of Student-t distributions. Secondly, the mean estimator is preferred because of the reconstruction quality while depending on less coefficients compared to the median and mode estimators. Finally, experiments validate the use of constant chroma regressors per kernel, which results in smoothed piecewise constant chroma planes.

In contrast to traditional image and video representations, the presented Steered Mixture-of-Experts representation scales easily to higher dimensional image modalities. In the next chapter, the approach is applied to higher-dimensional immersive image modalities.

4

SMoE for Immersive Image Modalities

4.1 Introduction

In this chapter, the focus lies on image modalities that provide some form of immersive experience. As shown in Chapter 1, the ultimate goal is to arrive at a representation that allows for 6-DoF, i.e. allowing the user to move around and to look around as they please. 360-degree images and video were the first step in that direction. The viewer is able to look around, but the viewpoint is fixed in space. As shown in Chapter 2, all required visual information for a full 6-DoF experience can be mathematically expressed as the plenoptic function. When the "open space" assumption is met, then the plenoptic function can be represented by the 4-D light field. Light field images and light field videos enable the desired 6-DoF functionality over a space where no objects are present.

In Chapter 3, the unifying SMoE framework for representing image modalities was presented. Remarkably, the representation scales to any dimensionality, which is in stark contrast to traditional coding paradigms. However, some challenges are still present when scaling to new image modalities. This chapter thus focuses on how the representation is adapted to immersive image modalities and focuses on some of the functionalities specific to these modalities and how this translates to the SMoE representation.

This chapter first focuses on spherical image dimensions. The previous chapter only considered Euclidean coordinate spaces. However, spherical dimensions are often present in immersive 6-DoF application, as shown in Chapter 2. Exam-

ples include 360-degree images and video, spherical light fields and the plenoptic function that contains two spherical coordinate dimensions. Secondly, the SMoE representation is applied on 4-D light field images. Finally, the representation is extended to 5-D light field video.

Note that a secondary challenge with these immersive image modalities is the enormous sets of samples. Fitting mixture models on such large datasets is a big challenge by itself. Therefore, Chapter 5 is dedicated to discussing how the EM modeling can be optimized for such datasets. In this chapter, the focus is put on the properties and approximation capabilities of SMoE models for these immersive image modalities.

4.2 SMoE for Spherical Coordinate Dimensions

In Chapter 3, it was shown how to build a Mixture-of-Experts based on mixture models. In order to enable spherical dimensions, one option could thus be to adopt mixture models from the field of directional statistics. Multivariate versions of directional distributions exist in directional statistics, such as the *von Mises-Fisher* (vMF) and *Kent* distributions [100]. The vMF distribution is analogous to the symmetric Gaussian distribution, and thus cannot be steered. It was later generalized towards the steerable Fisher-Bingham distribution. Nevertheless, this steerable distribution is considered to be mathematically inelegant and lacks a natural interpretation of the parameters [101]. Kent suggested an alternative with more interpretable parameters and which is more flexible than the vMF distribution [102]. However, the normalization constant is not solvable in closed-form and the approximation of the constant is not always applicable [101]. Furthermore, the parameters are still less flexible and interpretable compared to the multivariate Gaussian. Fitting mixtures of Kent distributions has been proposed, but relies on the approximate normalization constant and remains computationally complex [103]. From these observations, I investigated the possibility to work with the mathematically elegant GMMs for SMoE on the unit sphere. This would also allow us to easily mix Euclidean and spherical dimensions within the same image modality, e.g. in the plenoptic function.

In this section, a method is proposed to extend the SMoE framework to spherical dimensions based on the mathematically elegant GMMs [55]. As such, the SMoE representation could be used for approximating image modalities such as 360-degree images, spherical light fields, and even the entire plenoptic function in the future. Furthermore, a method is proposed to reduce the parameter space to the same two dimensional Euclidean space as for planar 2-D images by using a projection of the covariance matrices onto tangent spaces perpendicular to the unit sphere.

4.2.1 SMoE on the Unit Sphere

The goal is to develop a method for SMoE on spherical image data. Such image data has some specific properties that can be exploited. Firstly, the samples are densely and relatively uniformly distributed on the unit sphere, e.g. there are no regions where no samples are present. Secondly, in SMoE it is typical to work with mixture models with a high number of kernels that have small spatial variance. Consequentially, the assumption is that the unit sphere can be approximated by the kernel tangent planes, analogous to a circle that is approximated by a polygon with a high number of edges.

Based on the above observations, the spherical data is chosen to be interpreted as data with a 3-D coordinate laying on the unit sphere. The data is consequently modeled using a GMM with a 3-D Euclidean coordinate space. However, it is clear that this is a redundant parametrization as all the data lay on a manifold, i.e. the unit sphere. Later in the section, it is shown that the GMM can be projected onto a 2-D coordinate space by locally projecting each kernel's covariance matrix onto the tangent plane defined by the kernel center. The proposed idea is implemented for omnidirectional (360°) images, thus having two spherical coordinate dimensions and three Euclidean color dimensions, i.e. RGB.

In order to perform SMoE on samples with two spherical dimensions, each coordinate is first translated into a 3-D unit vector. Consequently, modeling can then be performed in a Cartesian space in which SMoE can straightforwardly be applied. The GMM is used to model the joint probability of the 3-D coordinate and 3-D color space, the model thus contains 6-D Gaussian kernels. Fig. 4.1 illustrates a SMoE model trained on image data laying on the unit sphere.

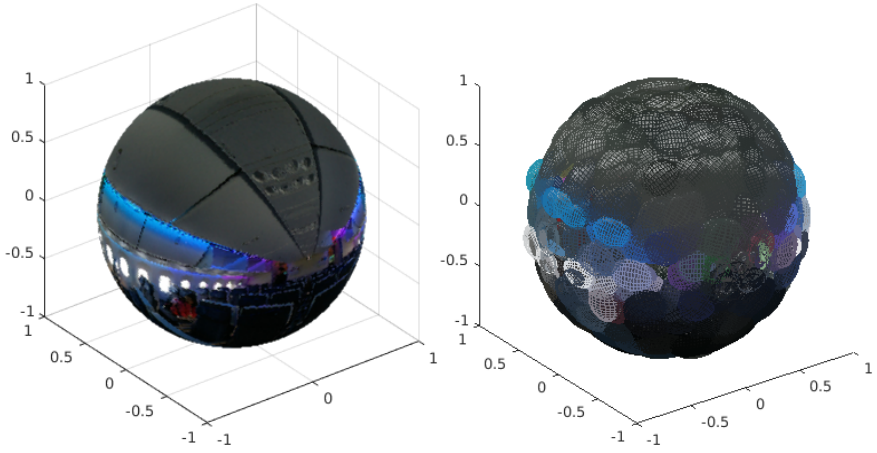
4.2.2 Projection onto the tangent space

As mentioned in Chapter 3, a kernel is defined by its prior π, center $\boldsymbol{\mu} = [\boldsymbol{\mu}_X; \boldsymbol{\mu}_Y]$ and covariance matrix R. In the case of kernels laying on the unit sphere, the coordinate center $\boldsymbol{\mu}_X \in \mathbb{R}^3$ can be seen as a vector radiating out from the center of the unit sphere. The subspace orthogonal to this vector, at the surface of this sphere, is necessarily tangential to the sphere and is given by $P_\perp = I - P$, where $P = \boldsymbol{\mu}_X \boldsymbol{\mu}_X^T / (\boldsymbol{\mu}_X^T \boldsymbol{\mu}_X)$ [106]. The coordinate space covariance matrix R_{XX} is approximated by projecting the covariance matrix onto $P_\perp$ as follows

$$R_{XX} \approx P_\perp R_{XX} P_\perp + P R_{XX} P, \tag{4.1}$$

where the first term is a projection of the coordinate covariance matrix onto the 2-D tangent plane and the second term is the contribution along $\boldsymbol{\mu}_X$. The expression is an approximation as the full equation includes the spaces $P R_{XX} P_\perp$ and $P_\perp R_{XX} P$. In practice, these contributions are small enough to ignore. It will be demonstrated in Sec. 4.2.4 that it is sufficiently accurate for our purpose.

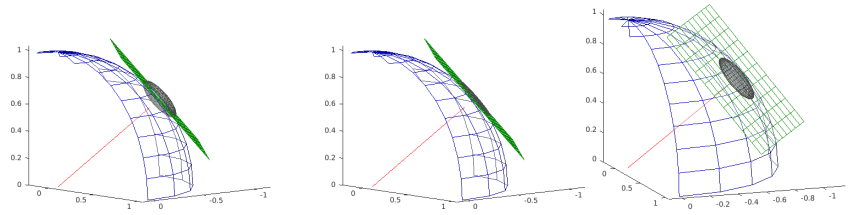
(a) Equirectangular projection (P10 [104], [105])



(b) Pixels on the sphere

(c) Approximated by GMM

Figure 4.1: Example of a SMoE model on the unit sphere without projection. Only the coordinate space is visible on the axes. The color space is visualized by the color of the ellipsoids.



(a) Original (b) Projected onto the tangent plane

Figure 4.2: Illustration of the projection of a single covariance matrix (a) onto the tangent plane $P^{\perp}$ (green plane). A small eigenvalue ϵ is added corresponding to the eigenvector that is defined by the coordinate center μ_X (red vector).

Observe that the unit sphere is infinitely thin. Therefore, when K is high enough, the contribution along $\boldsymbol{\mu}_X$ will become infinitely small. In this case, the covariance matrix R_{XX} is completely defined by the two eigenvectors (and corresponding eigenvalues), that lay in the 2-D plane $P_{\perp}$ and the third eigenvector is along the $\boldsymbol{\mu}_X$ direction with a small eigenvalue. This small eigenvalue can be fixed to a small scalar ϵ. The second term in Eq. 4.1 is thus approximated by ϵP.

Let us define the projected covariance matrix $\tilde{R}$ as being constructed by four submatrices analogously to Eq. 3.10:

$$\tilde{R}_{XX} = P_{\perp} R_{XX} P_{\perp} \tag{4.2}$$

$$\tilde{R}_{XY} = \tilde{R}_{YX}^{T} = R_{XY} P_{\perp} \tag{4.3}$$

$$\tilde{R}_{YY} = R_{YY}, \tag{4.4}$$

with $\tilde{R}_{XX}$ and $\tilde{R}_{XY}$ now being of rank-2.

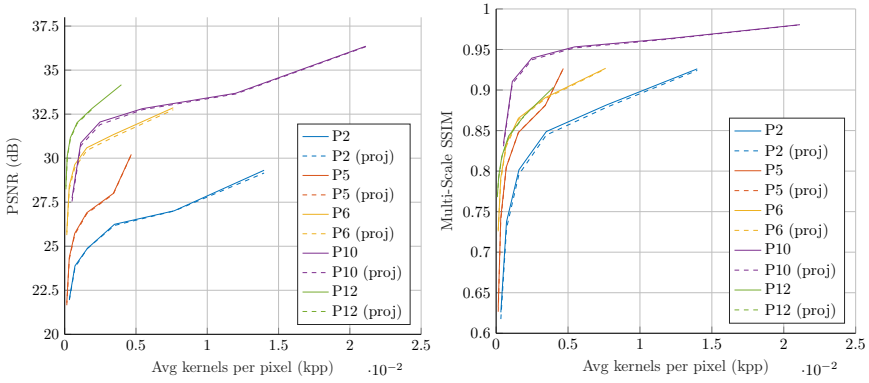
4.2.3 Dimensionality reduction

In this section, it is shown that it is possible to parametrize the two spherical dimensions with the same number of parameters as two Euclidean dimensions (planar images), i.e. having a $\dot{\boldsymbol{\mu}}_X \in \mathbb{R}^2$, $\dot{R}_{XX} \in \mathbb{R}^{2\times 2}$ and thus $\dot{\boldsymbol{\mu}} \in \mathbb{R}^5$, $\dot{R} \in \mathbb{R}^{5\times 5}$.

The coordinate center $\boldsymbol{\mu}_X$ approximates a unit vector when K goes to infinity as it is the mean of an ever decreasing amount of data laying on a small segment of the sphere. It can therefore be parametrized by two coefficients as the norm is one, i.e. $\dot{\boldsymbol{\mu}}_X \in \mathbb{R}^2$. Using the eigenvalue decomposition, the following can be shown

$$\tilde{R}_{XX} = UDU^T = [\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3] \begin{bmatrix} d_1 & & \\ & d_2 & \\ & & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^T \\ \boldsymbol{u}_2^T \\ \boldsymbol{u}_3^T \end{bmatrix} \tag{4.5}$$

$$= d_1 \boldsymbol{u}_1 \boldsymbol{u}_1^T + d_2 \boldsymbol{u}_2 \boldsymbol{u}_2^T \tag{4.6}$$



(a) PSNR (b) Multi-Scale SSIM

Figure 4.3: Experiment results comparing the modeling with and without the projection in terms of PSNR and Multi-Scale SSIM.

Let $\dot{R}_{XX}$ be the desired 2×2 covariance that defines the covariance in the 2-D $P_\perp$ plane. It is then possible to construct $\dot{R}_{XX}$ by taking the top-left four elements of $\tilde{R}_{XX}$:

$$\tilde{R}_{XX} = \begin{bmatrix} \dot{R}_{XX} & \begin{bmatrix} a \\ b \end{bmatrix} \\ \begin{bmatrix} a & b \end{bmatrix} & c \end{bmatrix}. \tag{4.7}$$

At decoder side, the variables a, b, and c can be found by solving $P\tilde{R}_{XX} = 0$ using linear operations. Finally, in order to have a small positive eigenvalue ϵ along $\boldsymbol{\mu}_X$, ϵP is added to $\tilde{R}_{XX}$.

Note that the dimension reduction is analogous for $\tilde{R}_{XY}$. However, the eigenvalues for $\tilde{R}_{XY}$ are not altered, which means that there is no color gradient along $\boldsymbol{\mu}_X$ any longer. This is the information that is lost using this projection, however this is not critical since it is the color gradient along the line of sight.

4.2.4 Omnidirectional images

For the experiments, five images were selected from the *Salient360!* dataset: P2, P5, P6, P10, and P12 [104], [105]. These images were stored in equirectangular format. After remapping the pixels onto the unit sphere, these images are progressively modeled using minibatches of size 10,000 and local updates per 18°-by-18° segments (see Chapter 5). Models are initialized uniformly on the sphere with $K_{\text{max}} = 2^{12}$ kernels. After each meta-iteration, 40% of the kernels are split based on their weighted conditional variance. The modeling stops when $K_{\text{max}} = 2^{18}$ is reached.

Fig. 4.3 shows the objective quality results for the indicated images in terms of PSNR and *multi-scale structural similarity* (MS-SSIM) [107]. The x-axis is expressed in *kernels-per-pixel* (kpp), as the original resolutions of the images span from 2000×4000 (P10) to 5000×10000 (P12). Note that the initial number of kernels is fixed, each image thus spans a different kpp range. The plots are shown up to 0.02kpp, which indicates that on average one kernel spans 50 pixels in the original equirectangle image. For P10 (Fig. 4.1) a 0.9 on the MS-SSIM scale is achieved at 0.001kpp, which is an average of 1 kernel per 1000 pixels. The average loss over all images is 0.1 dB PSNR and 0.002 MS-SSIM. Therefore, it can be concluded that the projection step introduces a relatively small quality loss, which indicates that the assumptions made are valid. Note that kernels can become insignificant during the modeling. These kernels are consequently removed, which influences the shape of the plots.

For omnidirectional images or video, it is a very desired property to spent more detail on the horizon that is not trivial in other coding mechanisms. In traditional approaches, the 360° images are always projected onto a 2-D plane. The projection heavily stretches the image around the poles, leading to a substantial overhead as much more pixels are spent on the poles. Furthermore, the poles are in general of little interest as viewers tend to focus on the horizon. Fig. 4.4 illustrates two models of the P12 omnidirectional image of which the second model has roughly 20 times as many kernels as the first image. The image nicely illustrates the benefit of SMoE distributing its kernels evenly over the unit sphere. This is visible as more detail is being spent along the horizon compared to the poles. The fact that the SMoE model evenly distributes its kernels over the unit sphere leads to the illusion that in equirectangular projection that there is much more detail on the horizon compared to the poles. Furthermore, due to the adaptive splitting method, more kernels are spawn where more detail is needed.

4.2.5 Conclusion

A method was presented for applying SMoE to spherical dimensions by operating on the unit sphere in the Euclidean $\mathbb{R}^3$ space. A computationally cheap technique to reduce the parameter space to the same space as for planar 2-D images is introduced. This is done by projecting the covariance matrices onto the 2-D tangent space defined by the kernel's center. Finally, a computationally efficient modeling scheme that utilizes this projection step is presented. Experiments validate that the parameter space reduction introduces nearly no quality loss for the tested 360° dataset. As such, evidence is shown that SMoE can be applied efficiently to spherical dimensions as needed for approximating the 5-D plenoptic function in the future.

(a) $K = 9.013$, 30.12 dB PSNR, 0.829 SSIM

(b) $K = 198.802$, 34.14 dB PSNR, 0.860 SSIM

Figure 4.4: Two SMoE models of the P12 360° image [104], [105] reconstructed in equirectangular projection. SMoE operates on the unit sphere, as such, kernels are evenly distributed along the sphere. When using equirectangular projection, the poles of the sphere are heavily stretched. It is noticeable that the SMoE model has the desirable effect of spending more detail along the horizon compared to the poles.

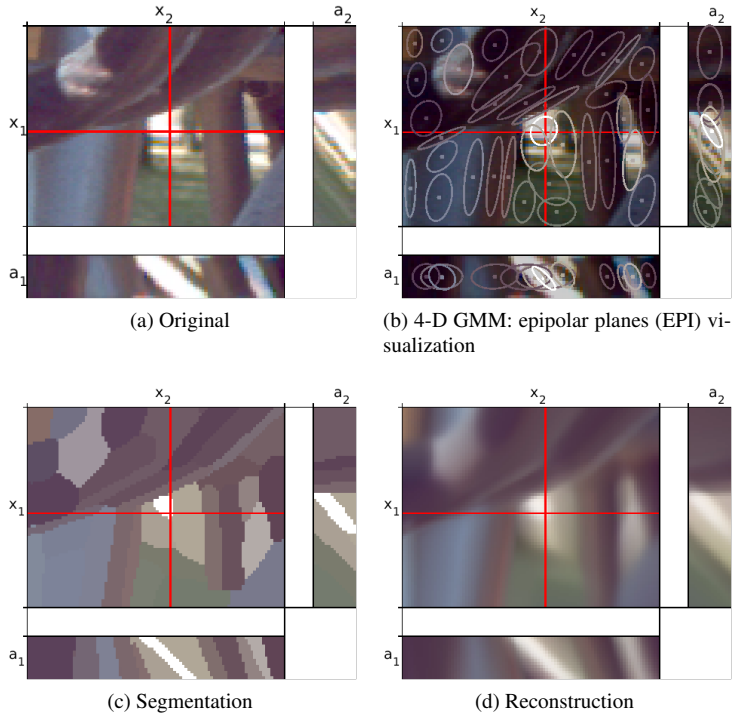
4.3 SMoE for Light Fields

In this section, the extension of SMoE towards static 4-D light fields is introduced. The 4-D light fields considered in this section are short-baseline LFs resulting from lenslet-type cameras. However, the theory does not rely on any hardware assumptions and is thus applicable to light fields from any acquisition source. As mentioned in Chapter 2, the following LF parametrization is chosen: $\mathrm{LF}(a_1, a_2, x_1, x_2) = (\mathrm{Y}, \mathrm{Cb}, \mathrm{Cr})$, in which (a_1, a_2) are the camera (row, column)-coordinates on the camera plane and (x_1, x_2) are the pixel (row, column)-coordinates on the image sensor. This parametrization is conform with the data structure that is yielded by the LF Toolbox v0.4 [23]. Consequently, the GMM is 7-D, with the X-coordinate being 4-D and the Y-amplitude being 3-D. Additionally, the soft-windows $w_j(\boldsymbol{x})$ (Eq. 3.16), describe a 4-D volume per kernel, and the expert function $m_j(\boldsymbol{x})$ (Eq. 3.14) describes for each color channel a 4-D gradient, i.e. a linear function from $\mathbb{R}^4$ to $\mathbb{R}$.

Fig. 4.5a shows a small light field, including the EPIs on the bottom and right side. The red lines indicate where the 4-D space is sliced, i.e. indicating where the EPIs are located spatially. As the kernels are likelihood optimized, they are expected to steer along the diagonal EPI structures. As such, kernels can be responsible for different pixel coordinates (x_1, x_2) depending on the camera coordinate (a_1, a_2). Visually, it seems thus that kernel windows move over the image plane when moving the viewpoint. The magnitudes of these shifts correspond to the slopes within the EPIs. Interestingly, these slopes are proportional to the depth of that point in the scene [110]. The orientation of the kernels along the EPIs thus implicitly codes depth and could potentially be used as a depth estimator as shown in Sec. 4.3.2 [57]. Furthermore, a single kernel can yield different color values when viewed from a different camera coordinate through the 4-D gradient. As such, it allows us to model non-Lambertian reflectance.

Fig. 4.5b shows a low order GMM fit onto the data, note how the kernels have a spread in all four coordinate dimensions simultaneously. Fig. 4.5c illustrates the segmentation, which is nothing more than the hard-decision of the soft-windows $w_j(\boldsymbol{x})$. It is clear that the windows steer along the EPI structure and soft-partition the entire 4-D space, thus yielding global support. Using Eq. 3.17, the reconstruction is illustrated in Fig. 4.5d. Next, in order to better visualize that the kernels have a volume in coordinate space, a projection to the 3-D subspace (a_1, x_1, x_2) (projecting vertically along the camera plane) is shown in Fig. 4.6a and 4.6b. Finally, Fig. 4.7 illustrates the kernels for another light field for illustration purposes.

Fig. 4.8b shows the reconstructed (7,7)-view from the *I01 Bikes* light field shown in Fig. 4.8a [108]. For more details on the modeling, the reader is referred to Chapter 5. Note how the speckle rust turns into smudges in the reconstruction, which could arguably be seen as a visually-pleasing quality decay. This is however



(a) Original

(b) 4-D GMM: epipolar planes (EPI) visualization

(c) Segmentation

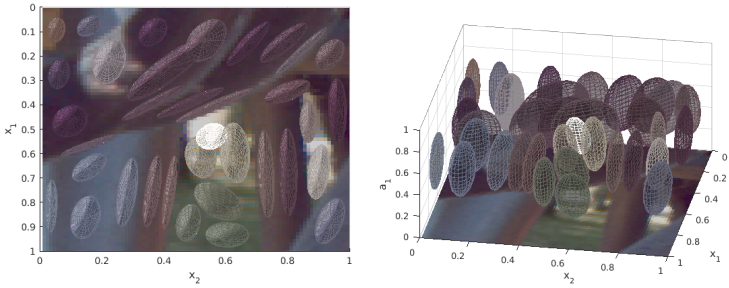
(d) Reconstruction

*Figure 4.5: SMoE modeling and reconstruction of a cropped LF (*I01 Bikes *[108][109]) using a very low amount of kernels for visualization purposes (K=35). The original is shown in (a). The kernels are visualized in the 4-D coordinate space with camera coordinate dimensions (a_1, a_2) and pixel coordinate dimensions (x_1, x_2) in (b). The EPI images are shown below and right of the spatial crop, corresponding to the pixels indicated by the red lines. The reconstruction is shown in (d). Note that SMoE implicitly provides a 4-D consistent segmentation (c), when is indicated for each pixel $\boldsymbol{x}$ by the kernel j who is dominant (highest $w_j(\boldsymbol{x})$). The segmentation illustrates how the kernels are steered along the EPI diagonal lines.*

heavily penalized when using objective metrics such as PSNR and SSIM [98]. Note that, the reconstruction is slightly blurred due to the relatively low number of components ($K = 8960$) compared to 41.483.904 original pixels in the lenslet image. Thus resulting in 4.630 pixels for one component on average, i.e. each 4-D soft-window spans 4.630 samples on average.

4.3.1 View interpolation

Important to note is that the method is able to reconstruct views that were not captured, which is in stark contrast to dense representations that require a separate



(a) 3-D (a_1, x_1, x_2) projection: frontal view

(b) 3-D (a_1, x_1, x_2) projection: angled view

Figure 4.6: In order to illustrate that the kernels in Fig. 4.5 represent a volume, the kernels are projected on the 3-D space along the dimension a_2 (thus ignoring horizontal parallax). As such, it is clear that in this 3-D space, the kernels form ellipsoids as can be seen in (a) and (b). The center of the kernels in the color dimensions, i.e. $\boldsymbol{\mu}_Y$ is shown as the color of the ellipsoid.

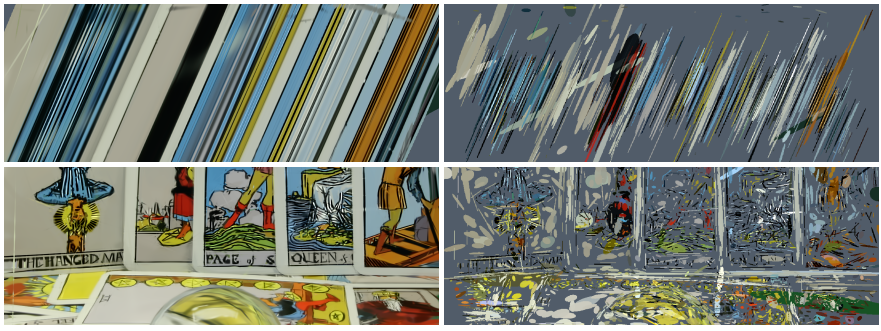


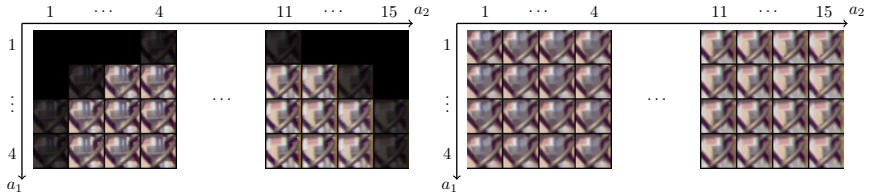
Figure 4.7: Illustration of SMoE kernels of a 4-D light field (original: Stanford Light Field Archive [111]). The reconstruction (left) and the employed continuous kernels are shown using an artificial cutoff for visualization purposes (right). The top shows the horizontal EPI, whereas the bottom shows the spatial view.

view synthesis process. The SMoE model has a continuous representation, as such any view in the domain can be readily reconstructed. Moreover, limited extrapolation is also possible. Fig. 4.9a, shows that the LF data structure (obtained through the MATLAB LF Toolbox [23]) results in black views in the corner directional views. The SMoE method is able to estimate these views with remarkable consistency by excluding the black views during training. The effect is clearly visible by the position of the red square on the background (Fig. 4.9b). However, extensive evaluation is considered as future work.

(a) Original view

(b) Reconstruction

Figure 4.8: I01 Bikes *[108][109] light field example (K=8960), showing a central view with* $(a_1, a_2) = (7, 7)$ *with mean* $PSNR_{YCbCr}$*: 30.71 dB and mean* $SSIM_Y$*: 0.86 (objective evaluation as in [108]). Note the smoothing artifacts in (b) which originates from kernels being responsible for a large number of pixels. For example, the mud speckles on the "peugeot" bar in (a) turn into smudges in (b).*



(a) Original views (b) Reconstructed views

Figure 4.9: In the original light field there are missing views due to the sensor architecture. Views that are at the outer edges of the camera plane are shown in black, with a_1 and a_2 corresponding to row and column in the camera plane. However, view reconstruction using SMoE (26×27 *spatial crop from* I02*) shows consistent extrapolation of these outer views.*

4.3.2 Depth estimation

In Sec. 2.3.3, it was explained that 4-D light fields implicitly include depth information. The slopes of the lines in the EPI plane are related to the depth of the corresponding points in the 3-D space [112]. Similar to the work of [113], continuous depth values can be obtained by estimating the slope of lines in EPIs. Interestingly, Fig. 4.5 showed that the SMoE kernels are steered along the EPI strips. The orientation of the kernels over the EPI planes thus implicitly corresponds to depth.

The slopes in the EPIs correspond to the kernel's covariance between the spatial dimensions and the camera plane dimensions. However, this is expressed per kernel. In order to have a full depth map for each pixel, a weighted sum is performed over these kernel-depth values, analogous to the regression (Eq. 3.17). For each pixel, the slope in the EPI planes is determined by the weighted sum of the covariances between the camera plane and spatial dimensions. This results in two depth estimations: one corresponding to horizontal camera movement, and one to vertical camera movement. For each kernel, the horizontal kernel EPI angle α_{H} is defined as follows:

$$\alpha_{\mathrm{H}} = \mathrm{atan2}(e_{\mathrm{H}}) \tag{4.8}$$

with e_{H} being the largest eigenvector of the 2-D covariance matrix of a_1 and x_1. Analogously for α_{V} with the covariance matrix of a_2 and x_2. In theory, there is no reason for the largest eigenvector to be the one corresponding to the lines on the EPI or the vector perpendicular to this line. However, in practice, this seems to be the case because the variance in the camera plane is typically larger than in the spatial dimensions.

To obtain a continuous angle value for a pixel i, the same weighted sum can be

used as for the regression:

$$\alpha_{i,\mathrm{H}} = \sum_{j=1}^{K} w_j(x_i)\alpha_{\mathrm{H}}^{j} \tag{4.9}$$

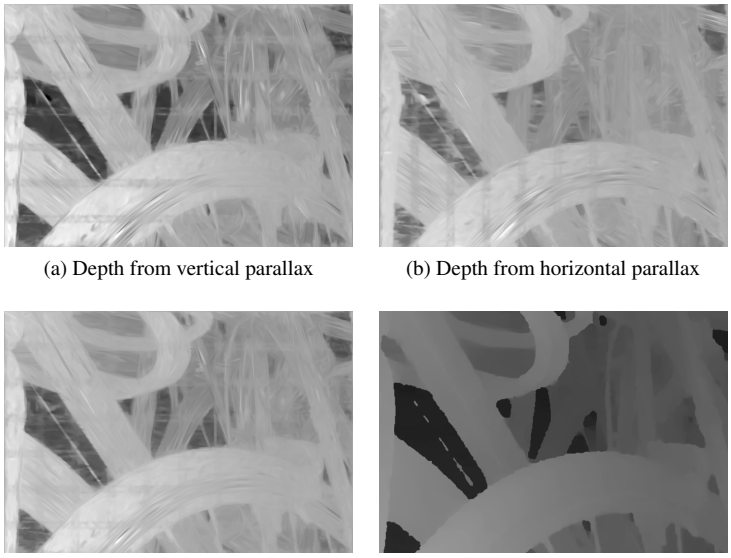
Results are shown in Fig. 4.10, in which darker parts are further away from the camera. It is clear that this simple approach is able to perform an interesting depth estimation. The estimation is noisy, but is continuous and detailed, compared to the depth map in Fig. 4.10d provided by the Lytro software [114]. Note the very narrow brake cable under the "Peugeot" beam. The vertical and horizontal block artifacts are caused by a block-wise modeling approach, detailed in Chapter 5 (blocksize=128).

In Sec. 2.3.3, it was noted that there are typically two stages in depth estimation based on light fields: (1) a rough depth estimation, and (2) a depth refinement stage. The initial depth map typically contains outliers due to noise, occlusions, or inherent matching uncertainty cased by textureless regions [20]. In SMoE, the inherent matching uncertainty is the largest reason for outliers. Kernels steer along the correlation of pixel amplitudes. As such, if a larger region has no texture, then the kernels are not steered along the EPI strips. In such a case, the kernel is oriented well in terms of likelihood optimization, but does not correspond to the EPI strips no more. Therefore, a depth refinement stage would be advised here as well.

4.3.3 Edge detection and other descriptors

As shown in Sec. 3.4.2, edge detection can be derived from the model parameters without having to reconstruct the views. Each kernel describes an edge which is described by the norm of $R_{YX,j}R_{XX,j}^{-1}$. The same can be done for 4-D LFs. Consequently, not only are the spatial gradients taken into account, but also the variation of the luma along the camera place dimensions. Fig. 4.11 shows an example comparison of using only the spatial dimensions for edge detection (x_1 and x_2), and the situation where the full covariance is used (a_1, a_2, x_1, and x_2). 4-D edge detection results in a much more detailed result as information is used from multiple viewing angles. Interestingly, using only the 2-D gradients in Fig. 4.11 fails to capture the same number of edges as, e.g. the Sobol edge detector which also only depends on the spatial gradients. This can be explained by the fact that edges can be represented in two ways in the model: (1) as the luma gradient expressed by the covariance between spatial dimensions and luma dimension, and (2) as the transition of one kernel into another which is the result of the softmax weighing. Only the first type of edges are exposed using the described technique.

Sec. 3.4.2 also described segmentation based on the kernel parameters. Fig. 4.5 illustrates the segmentation of a small light field patch. A 4-D segmentation results when selecting the most dominant kernel for each pixel. Due to the high number of



(a) Depth from vertical parallax

(b) Depth from horizontal parallax

(c) Mean depth

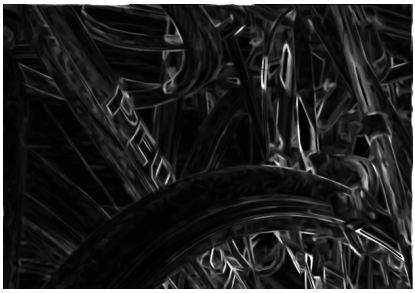
(d) Depth map from Lytro software [114]

Figure 4.10: Depth estimation based on the model from Fig. 4.8. It is known that the slopes in EPIs are linked to the depth of that pixel in the scene. The kernels are steered along these EPI slopes, and thus the parameters of each kernel capture this slope in a continuous manner. Rough depth can thus be estimated without reconstructing the view and is thus available in the compressed domain.

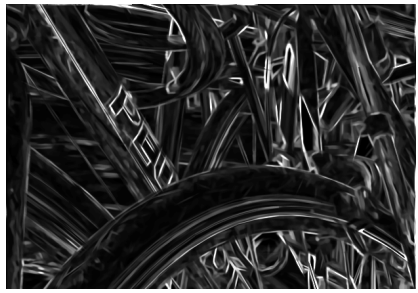
samples in a light field, it is very beneficial to have a consistent 4-D segmentation. Recent work proposed the extension of the concept of superpixels to light fields, i.e. superrays [51]. The intention of the work is to have these clusters as light field atoms to allow for efficient processing of light fields, very similar to the SMoE approach. This dissertation focuses on the SMoE applications of approximation and coding, extensive evaluation of these other applications are considered future work.

4.3.4 Pixel-parallel real-time view reconstruction

Recall that each pixel is independently reconstructed. Furthermore, due to the modeling methods used in Chapter 5, each pixel is only dependent of kernels in its vicinity. This allows us to reconstruct each pixel in parallel using a limited number of kernels-per-pixel. It was shown that pixel-parallel reconstruction in real-time is possible given appropriate hardware using an OpenCL implementation [18]. The OpenCL implementation can achieve 85fps and 22fps for respectively 1080p and 4K renderings of large models with more than 100.000 of Gaussian kernels.



(a) SMoE 2-D edge detection



(b) SMoE 4-D edge detection

(c) Sobol edge detection from original

(d) Sobol edge detection on SMoE reconstruction

Figure 4.11: This figure illustrates the edge detection capabilities within the compressed domain using only parameters from the model in Fig. 4.8. Top left shows when only the kernel gradient magnitude of the spatial dimensions are used, whereas top right shows the improved edge detection when using the norm of the entire 4-D gradient. Bottom left shows the Sobol edge detection on the original (7,7)-view. Bottom right illustrates the Sobol edge detection on the SMoE reconstructed view.

This is a remarkable feature as the serial nature of these old paradigms (e.g., intra-prediction) makes it impossible to really achieve pixel-level parallelism. The current trend in hardware processing is a large increase of the number of execution threads, especially on *graphics processing units* (GPUs). The speed of a single thread is not being increased as fast anymore compared to a decade ago. Nevertheless, some parallelism is pursued based on traditional video coding standards such as HEVC, and is based on smart implementations such as the wavefront approach [17]. This ensures that blocks are decoded as soon as their dependencies are available. However, using 64-by-64 CTU blocks in a 1080p video only allows for only 15 decoding blocks. In the case of 32-by-32 CTU blocks, one can achieve 30 parallel streams. Such a scheme does fit multi-threading architectures, but is less suited for massively parallel architectures. However, this disseration focuses on the approximation and coding aspect of SMoE.

4.3.5 SMoE in a light field processing pipeline

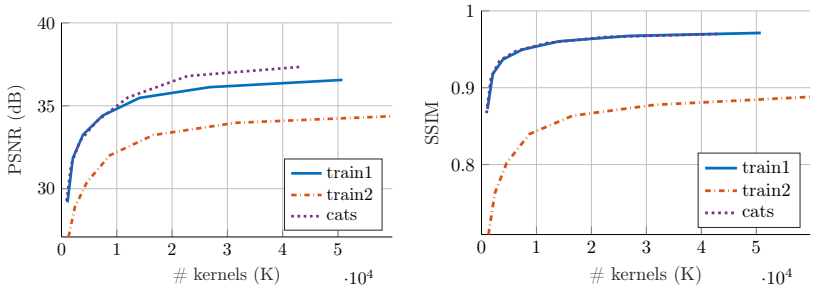
Remember Fig. 2.7 from Chapter 2. This figure by Wu et al. shows an overview of typical LF-pipeline tasks, divided into three categories: (1) low-level acquisition, (2) mid-level processing (super-resolution, depth estimation, compression), and (3) high-level user interface (application, editing, displaying) [20]. Based on the discussion of the properties of SMoE for light field in the previous subsections, one could argue that a representation such as SMoE is an adequate representation for the entire pipeline after the acquisition.

As shown in this section, once the camera data is modeled by SMoE, then mid-level processing such as resampling, and depth-estimation and compression can all be derived from the SMoE model. Similarly, for high-level user interface applications, rendering can be derived straight from the model. This shows that the geometrical interpretation of the SMoE parameters could facilitate such future applications. In this dissertation, the in-depth evaluation of these secondary functionalities was considered out of scope and there is therefore no guarantee that this would outperform the state-of-the-art. Nevertheless, these illustrations provide early indications that sparse models such as SMoE could be practical in a LF production pipeline as all subtasks could be derived straight from the model parameters.

4.4 SMoE for Light Field Video

In this section, the dimensionality is further increased by adding the time dimension to the 4-D LF. Light field videos are light fields are captured at intervals over time and thus yield a 5-D coordinate space (t, a_1, a_2, x_1, x_2). Not much is different in terms of theory compared to 4-D LF images, only now all kernels possess a time dimension. However, the time dimension does behave very differently compared to the camera-plane dimensions and the spatial dimensions. In practice, the kernels are all elongated along the EPI strips and have rather limited spatial range. However, along the time dimension both are commonly present. Kernels that represent the light irradiated by a static or a linearly-moving object will be spread long along the time dimension. However, in the case of non-linear movement or rapidly changing color values, kernels will be short along the time dimension. As shown in Chapter 5, this needs to be taken into account during modeling as an adequate kernel density over the whole 5-D coordinate space is desired.

In most cases, the frames for each viewpoint are synchronized during acquisition or are resampled before processing. However, as the SMoE representation is resolution agnostic, frames can be captured at irregular intervals. Only the absolute timestamp t of the frame is necessary for the modeling process. As the model is continuous, all views can be reconstructed at synchronous timestamps without



(a) PSNR (b) SSIM

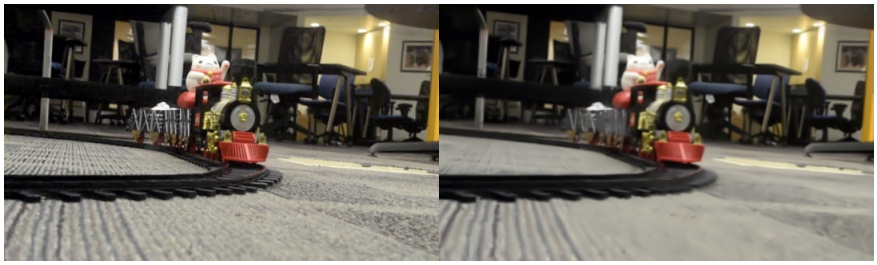
Figure 4.12: This figure illustrates the objective quality results in terms of PSNR and SSIM for the three light field videos [115].

any other methods involved.

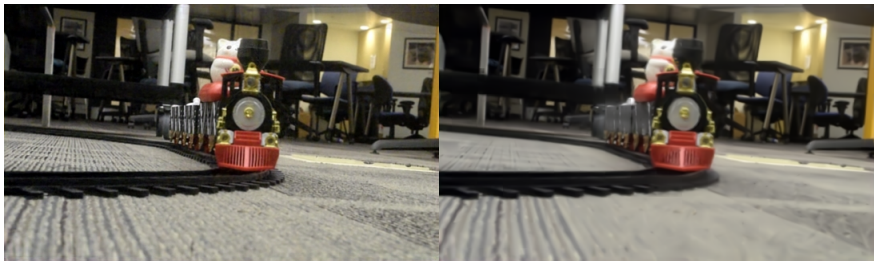
The main challenge in light field video is again an incredible number of samples that need to be modeled. A light field with 10×10 viewpoints, in full HD at 30 frames-per-second thus yields 6.220.800.000 pixels per second! Especially for these higher-dimensional modalities, sparse representations such as SMoE are hugely beneficial as a single kernel can span over a large number of pixels spread out over five dimensions simultaneously. In Chapter 5, an in-depth analysis is provided of the modeling process for such huge datasets.

Fig. 4.13 shows the short *train2* light field video [115]. The original video was captured using two cameras: a Lytro lenslet-camera that recorded at 3 frames-per-second, and a DSLR camera at 30 frames-per-second. Wang et al. then cleverly combine these two videos to produce one LF video of 30 frames-per-second (FPS) [115]. However, this process did introduce some artefacts due to the temporal upsampling. In general, this video is well reconstructed by SMoE and provides smooth linear motion of the train without much temporal issues. The main issues remain the fact that the fine texture of the carpet is not well approximated by a smoothed piecewise linear function. The most remarkable observation is that the necessary number of kernels does not grow exponentially compared to static light fields, while the number of original samples do. The kernels of the model in Fig. 4.13 cover 18.000 pixels on average! Fig. 4.12 further illustrates the relation between the number of kernels and the achieved objective quality for the three light field videos.

Fig. 4.14 illustrates two SMoE models of the *cats* light field video [115]. This video is interesting as the exhibited motion is periodic. The closest cat swings from left to right as a pendulum. Also the arms of both cats have the same swinging behavior but with a different period length. Because of this periodic behavior the modeling process can be fooled to find correlation over time as the pixels return



(a) Frame 3



(b) Frame 37

(c) Frame 95

Figure 4.13: This figure illustrates a three different views (a_1, a_2) at different frames t of the light field video train2 *[115]. The left is the original light field video which contains 1.080.688.640 pixels, i.e. 8x8 views, 97 frames, with a resolution of 320x544 pixels. The right is the SMoE reconstruction using $K = 59.827$ kernels, i.e. each kernel covers ± 18.000 original pixels! However, it is clear that with this number of kernels, some texture is disappearing and blurring is visible.*

Figure 4.14: This figure illustrates view $(a_1, a_2) = (7, 7)$ *at different frames* $t = (1, 55, 109)$ *of two SMoE models of the same light field video* cats *[115]. In this video, the right cat moves left to right as a pendulum, as well as the arms of both cats. The left model was modeled using* $K = 6.319$ *kernels, which resulted in 34.05 dB PSNR and 0.935 SSIM. The right model consists of* 43.334 *kernels and resulted in 37.35 dB PSNR and 0.964 SSIM. It is clear that the largest static part of the video is very well reconstructed which explains the rather high objective quality results. However, it is clear that the closest figurine suffers from strong temporal ghosting when the number of kernels is low.*

to their original (x_1, x_2) location after a certain time. This results in the observed temporal ghosting effect. More kernels or more intelligent modeling could mitigate this problem.

The same descriptors as for SMoE light field models are available (e.g. depth and segmentation), however, now for each point in time. Furthermore, the orientation of the kernels are likely to follow motion as the modeling process steers the kernels to harvest correlation over time. Extensive evaluation is considered future work.

4.5 Conclusion

In this chapter, the application of SMoE on higher-dimensional image modalities that target more immersive visual experiences was investigated. First, it was shown that spherical dimensions can be modeled by operating on the unit sphere. Furthermore, it was shown that the model can be parameterized in a per-kernel 2-D space by projecting each kernel onto the tangent space for that kernel. As such, working on spherical dimensions requires equal number of parameters as working in Euclidean spaces. Secondly, the extension of SMoE to light fields was discussed which revealed that SMoE models can adequately model light fields while being extremely functional and descriptive. Finally, the extension to light field video was introduced and discussed.

This chapter thus highlighted one of the key features of SMoE, i.e. dimension scalability. Dense representations have the problem that the number of samples grow exponentially in terms of dimensions. SMoE models have a relation that is more linear. The illustrative examples in this chapter resulted in an average pixels-per-kernel (ppk) of ± 100 ppk for 2-D and 360-degree images, ± 10.000 ppk for 4-D light fields, and ± 30.000 ppk for the light field videos. The exact relation between the number of pixels and the number of kernels necessary is hard to define as the number of kernels is highly dependent on the image content. A completely static light field video does not require more kernels than a light field image as kernels can span the entire time dimension. However, a light field video with a lot of non-linear motion does require a higher number of kernels.

In practice, the type of dimension clearly influences the average span of a kernel along that dimension in order to have a satisfactory reconstruction quality. In the results, the kernels typically range over a small number of pixels along the spatial dimensions and very high numbers of pixels along the camera dimensions in light fields. The kernel spread along the time dimension typically depends on the motion of the pixel patches. Strong non-linear motion requires a small spread along the time dimension, whereas static objects can span over the entire time dimension. This is very desired data-adaptive behavior where more bits are spent where there is more new information.

The main limitation remains to model high frequencies regardless of the type of dimension. Temporal flickering or spatial fine texture are not well approximated by a smoothed piecewise linear function. However, in theory we are not limited to this exact parametrization, and the model could thus be enriched to mitigate this limitation. Interestingly, the models have shown to also exhibit descriptive features for the newly added dimensions that are specific to some image modalities. The most striking descriptor is the depth estimation based purely on the model parameters in light field modalities without view reconstruction. This thus allows for many operations in the compressed domain. Future work will consist of further investigating and coupling the model parameters to the semantics of the description.

5 Building Mixture Models From Extremely Large Datasets

5.1 Introduction

During the previous chapters, the SMoE framework was proposed and applied to several image modalities while discussing the particular characteristics of the SMoE models for each image modality. From an engineering perspective, it is not trivial to build such models as the outlined EM algorithm in Sec. 3.3.2 scales poorly towards large datasets and large numbers of distributions. This chapter is dedicated to modeling mixture models of extremely large datasets by discussing the state of the art and our proposed modeling scheme for extremely large datasets.

In video compression standardization organizations, it is common to only standardize the decoder of compression scheme, or more precisely, the bitstream format. This allows for competitiveness in encoder development. Two encoders might produce the same bitrate but yield different quality levels. The same concept is available in SMoE. Two modeling schemes can produce the same number of kernels but yield wildly different quality. The modeling process must thus be approached with incredible care. To be precise, as will be discussed in the next chapter, the redundancy between the kernel parameters can also lead to different bitrates. However, this chapter focuses on the modeling, i.e. the fitting of the kernels onto the pixel data.

As shown in the previous chapter, the number of pixels to be modeled increases exponentially with the number of dimensions. The modeling thus becomes more

challenging when the dimensionality increases. The $\pm$ 3 second light field videos in the previous chapter thus contain roughly 1 billion pixels with 3 color channels yielding 2.86 GiB when stored in 8-bit unsigned integers or 11.4 GiB when using 32-bit floats. As such, it is nearly impossible to keep all data in memory during modeling. Furthermore, the basic EM algorithm requires $O(NK)$ operations and memory, with N being the number of pixels and K being the number of distributions. It is clear this is problematic when working with billions of samples and thousands of distributions.

Luckily, there are ways for us to reduce the complexity of the modeling process. Remember that in Sec. 3.3.1, the paradigm of conditional computing in MoEs was discussed [91]. The idea is that during evaluating or modeling not all branches of an MoE need to be activated because many branches will be zero-weighted. It is therefore important to first do the gating before evaluating all the branches. In SMoE the gating operates purely in the coordinate space and branches correspond to kernels. In other words, each pixel is dependent on only a small set of kernels and each kernel is typically constructed from a small set of pixels. This observation enables us to perform *local modeling*, in which each kernel is only aware of the set of pixels in its vicinity.

The chapter is structured as follows. First, the EM algorithm is discussed and an overview of the complexity-reduction methods in the literature is presented. Second, the approaches used in this dissertation are discussed that are tailored to the corresponding image modalities.

5.2 Insights and Complexity of EM

5.2.1 The EM algorithm and latent variables

The Expectation-Maximization algorithm is an iterative method that is used for maximizing the likelihood of the parameters of a model so that the model best explains the observed data. The main idea is that the model is dependent on a set of latent variables. Latent variables are unobserved entities that are not readily present in the observed data, but do link sets of the observed data together, comparable to categorizing data into a smaller number of sets. The goal of identifying latent variables is to cleverly group the data together in order to more easily process that data. In our case, our latent variables are the parameters of the Gaussian kernels that correspond to coherent regions of the plenoptic function. Depending on the coordinates and the color amplitudes of the pixels, the corresponding generative kernel is inferred.

The EM process thus tries to solve a a chicken-and-egg problem: it wants to discover what the underlying kernels are, and then also evaluate which pixels then correspond to that kernel. However, the kernels themselves are constructed by the

statistics of the pixels that belong to that kernel. Therefore, the EM-algorithm iteratively goes through the phases of pixel-labeling (E-step) and kernel-identification (M-step). At the start of the algorithm, an initial guess of the kernel parameters is done based on prior knowledge or at random.

The likelihood for mixture models of members of the exponential function was introduced in Sec. 3.3.2. In this chapter, the focus lies on the application of SMoE where we jointly model pixel coordinates $\boldsymbol{x}$ and amplitudes $\boldsymbol{y}$ using GMMs. From Eq. 3.4, the likelihood of a GMM is thus defined as:

$$l(\Theta|X,Y) = E[\log p(\boldsymbol{x},\boldsymbol{y}|\Theta)] \tag{5.1}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\log\sum_{j=1}^{K}\pi_j\mathcal{N}(\boldsymbol{x}_i,\boldsymbol{y}_i|\boldsymbol{\mu}_j,R_j). \tag{5.2}$$

At each iteration, the so-called E-step is performed in order to obtain an approximation of the unknown true membership of each pixel to each kernel. In other words, the likelihood of each pixel coordinate and amplitude pair $(\boldsymbol{x}_i,\boldsymbol{y}_i)$ belonging to the $p+q$ dimensional distribution j is evaluated. Note that the fitting is thus performed based on pixel coordinate and pixel amplitude simultaneously. As such, spatial and color correlation is taken into account. The soft-membership of each pixel is approximated by using the current kernel parameters as the following function:

$$\hat{z}_{ij} = \frac{\pi_j\mathcal{N}(\boldsymbol{x}_i,\boldsymbol{y}_i;\boldsymbol{\mu}_j,R_j)}{\sum_{l=1}^{K}\pi_l\mathcal{N}(\boldsymbol{x}_i,\boldsymbol{y}_i;\boldsymbol{\mu}_l,R_l)}. \tag{5.3}$$

Given the above notation, the probability density function of a multivariate Gaussian distribution $\mathcal{N}(\cdot)$ is given by

$$\mathcal{N}(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\mu},R) = \frac{1}{(\sqrt{2\pi})^{p+q}|R_j|}e^{-\frac{1}{2}\left(\begin{bmatrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{bmatrix}-\boldsymbol{\mu}\right)R^{-1}\left(\begin{bmatrix}\boldsymbol{x}_i\\ \boldsymbol{y}_i\end{bmatrix}-\boldsymbol{\mu}\right)^T}, \tag{5.4}$$

with $\boldsymbol{\mu} = \left[\begin{smallmatrix}\boldsymbol{\mu}_X\\ \boldsymbol{\mu}_Y\end{smallmatrix}\right], R = \left[\begin{smallmatrix}R_{XX} & R_{XY}\\ R_{YX} & R_{YY}\end{smallmatrix}\right]$.

Given the estimated soft-memberships, the Gaussian kernel parameters are re-

estimated (M-step):

$$\pi_j = \frac{1}{N} \sum_{i=1}^{N} \hat{z}_{ij} \tag{5.5}$$

$$\boldsymbol{\mu}_j = \frac{1}{\pi_j} \sum_{i=1}^{N} \hat{z}_{ij} \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix} \tag{5.6}$$

$$R_j = \frac{1}{\pi_j} \sum_{i=1}^{N} \hat{z}_{ij} \left(\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix} - \boldsymbol{\mu}_j \right) \left(\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix} - \boldsymbol{\mu}_j \right)^T \tag{5.7}$$

It is noteworthy that the E-step (Eq. 5.3) for each pixel is independently calculated per-pixel. The M-step is independently calculated per kernel, based on the pixels that the kernel has an influence on Eq. 5.7. Let us define that a kernel has an influence on a pixel if and only if $\hat{z}_{ij} > 0$.

5.2.2 Complexity

The classical batch EM requires $O(NK)$ operations per iteration with N samples, and K clusters. It is easy to see that this is not feasible for identifying high numbers of clusters on an extremely high number of samples. Consequently, researchers have tried to constrain the number of operations needed to fit a model onto large sets of data. Four main strategies were identified in the literature: (1) stimulate faster convergence so that fewer iterations are needed, (2) use fewer samples per iteration, (3) use fewer kernels per iteration, and (4) parallel and hardware implementations.

5.2.2.1 Faster convergence

Faster convergence leads to fewer iterations, which thus lowers the computational demand. For example, better initialization allows the EM algorithm to start closer to a desired solution, thus limiting the number of necessary iterations [116]. Other techniques augment the data to guide the modeling process in order to achieve faster convergence [117].

5.2.2.2 Fewer samples per iteration

The main idea here is to update the kernels more frequently, or independently from one another. In the most extreme case, kernels can be updated after seeing a single datum. These methods are called *online* learning methods [118]. These methods were first introduced for single sample updates, but are relatively easily expendable to a *minibatch* method [119]. These methods operate more coarsely than single sample updates, but more granularly than the batch version. Working

in minibatches has the advantage over single datum updates as it allows to make more use of efficient vectorization. Furthermore, it avoids local minima as follows. As the data changes in each iteration, so do the local optima. The implicit loss function in each iteration is an approximation of the global loss function.

Another way of splitting up samples is performed by labeling samples, e.g. using multi-resolution kd trees [120]. Or to use a cheap, approximate distance measure to efficiently divide the data into overlapping subsets, so called "canopies" [121]. Furthermore, it is also possible that not all data is equally important. As such, a method was proposed that classifies data samples into three categories: samples that can be safely discarded, samples that may be compressed, and samples that need to be retained in memory [122].

5.2.2.3 Fewer kernels per iteration

In order to reduce the complexity, one could also lower the number of kernels K per iteration. A first example are *greedy* EM algorithms [123][124]. These methods start with a single (or a low number) of kernels, and add components sequentially until a maximum number K of kernels is reached or if a desired stopping criterion is reached. These methods are thus also useful when the number of kernels is unknown a priori.

5.2.2.4 Parallel and hardware implementations

Although not decreasing the theoretical algorithmic complexity, it is possible to heavily improve the performance of the EM algorithm through efficient implementation. As can be seen in Eq. 5.3, each $\hat{z}_{ij}$ is determined independently. Such pointwise calculations can thus massively be distributed over thousands of cores in a GPU. This is clear in the number of papers on the topic, each with their own particularities as the frameworks and tools to program on GPUs are still very rapidly evolving [125]–[127]. One thing to notice here is that smart memory management here greatly affects the performance.

Wolfe et al. introduced a fully distributed EM over a set of computational devices for very large datasets [128]. The key idea here is that each computing node only interacts with parameters relevant to its data. This idea of data localization is to be central in the methods used for SMoE. Guo, Fu, and Luk proposed an FPGA-based, or fully-pipelined EM "engine" [129] of which they claim outperforms GPU implementations by a factor of 28. For this variant of the EM algorithm, they introduced a fixed-point arithmetic pdf evaluation kit. Some memory-constrained (buffer) systems for EM have been proposed as well, such as the scalable EM [122].

5.3 Methods used in SMoE

In the previous section, four main strategies were identified that reduce the complexity of the EM algorithm on large datasets. Luckily, in our application there is room for many heuristics, mainly due to the fact that the pixel data is heavily structured in sample grids. Furthermore, the conditional computing principle in MoEs enables us to exploit the structure of the data and the locality of our kernels (cfr. Sec. 3.3.1). This section presents the optimizations made to the EM algorithm during the course of this PhD.

5.3.1 Initialization

The optimization problem in the EM algorithm is unfortunately non-convex and converges to a local optimum [93]. Consequently, EM is sensitive towards the initialization of the parameters of the kernels, i.e. kernel priors, centers and covariance matrices. The goal of this subsection is to share some rules of thumb that were used during this work.

First, there are multiple possibilities for initializing the center values of the coordinates $\boldsymbol{\mu}_X$, trading off initial model likelihood and initialization complexity. For initialization, the structure of our data is important. All datasets discussed in the previous chapters, have samples uniformly distributed in the coordinate space. As such, an initialization that has a more-or-less uniform initial distribution is a good starting point. In the experiments, I found that using the Sobol quasi-random sampling is a convenient way to achieve a good uniform spread over an undefined number of dimensions [130]. Other quasi-random sampling methods are expected to achieve similar results. The advantage of using quasi-random sampling using the Sobol sequence is that you can request any integer number of starting positions in constant time. Choosing the centers data completely at random does not provide the same guarantees and the non-deterministic behavior makes the algorithm less tractable. The k-means++ algorithm is often used for initializing GMMs [116]. In my tests, k-means++ outperforms the Sobol sequence in terms of final likelihood after EM. However, k-means++ has the same high computational complexity of the EM algorithm. In order to mitigate this limitation, I found that it was good practice to use k-means++ on a subset of the samples, e.g. 5% of the original samples when feasible. Second, after the sample locations are determined, the sample amplitudes are initialized. I found that it is a good practice to then sample the nearest value to $\boldsymbol{\mu}_X$ within the image modality to determine the initial center amplitude $\boldsymbol{\mu}_Y$.

Third, good initial values for the covariance matrix are desired. I found it practical during our work to have models that are not too strongly overlapping as it often results in an overly blurred image as the EM-algorithm does not converge to a sharp solution. As a rule of thumb, I found that if you want to avoid excessive

blurring it is important to keep the values of the spatial dimensions on diagonal of the covariance matrices R_{XX} as small as possible as long as it does not endanger the numerical stability. However, when initializing dimensions of which we know that the kernels are highly likely to extend over larger areas, e.g. the camera plane dimensions in light fields across the EPI planes, that larger values (e.g. half the total length of samples on the dimension) are good choices. For simplicity, I choose to keep the covariances R_{XY} to zero, so to initialize with constant regressors instead of gradients. When initializing the $\boldsymbol{\mu}_Y$ by the values of existing samples, we know that there will be samples close the kernel center. Therefore, the initial value of the covariance in the color space R_{YY} is less important.

Finally, note that each kernel has identical R matrices and the samples are evenly distributed in the coordinate space. Consequently, a good choice for the priors π_j is $1/N$ indicating a uniform prior distribution as the kernels are likely to have influence over the same number of samples in the first iteration.

5.3.2 Regularization

The incoming data lays on sample grids which is potentially tricky to handle using EM as explained as follows. Imagine a one-pixel horizontal black line on a white background. If a kernel tries to fit the black line, it will become infinitely thin vertically. Consequently, the vertical variance becomes zero and the covariance matrix becomes singular and thus non-invertible. Regularization is performed to ensure that the Gaussian has a considerable spread along each of the dimensions, i.e. the covariance matrix has positive eigenvalues. Regularization can be as simple as adding a diagonal matrix with very small numbers to the covariance matrix.

Alternatively, note that pixels are actually integrated over time on a sensor of which the single light-sensors have a considerable surface. The convention is to position the pixel thus in the center of the integrated space-time volume. However, this observation can be used for regularization. The general idea is to add noise to the coordinates at runtime that randomly place the sample within the integrated space-time volume by adding a random value between $[-\delta_x/2, +\delta_x/2]$ with δ_x being the intersample distance. As such, the pixel values never overlap with adjacent pixels while the space between the pixel values is filled. It is clear that in our horizontal black line example, the pixel line will now have a variance along the vertical dimension at subpixel precision.

5.3.3 Block-based modeling

One divide-and-conquer based idea is to subdivide the EM modeling problem into smaller-sized subproblems. The easiest method in reducing the complexity of an image is to tile the image modality into p-dimensional blocks. A GMM is fit onto that block. Each block is then represented as a set of kernels. All submodels can

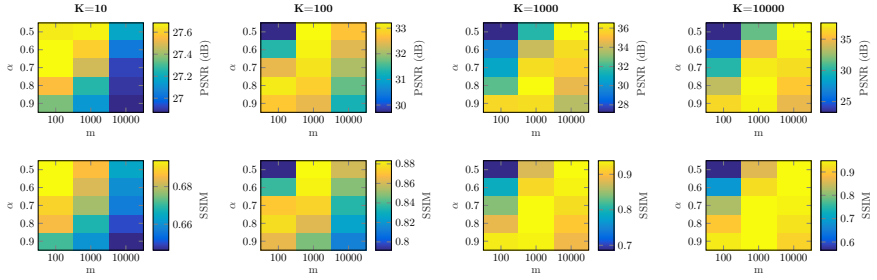


Figure 5.1: Analysis of the parameters α and the minibatch size m in terms of PSNR and SSIM performed on a large set of 324 small light fields. It can be seen that different number of kernels require different optimal (α, m)-pairs.

be merged into one large model that represent the entire image by adding a shift to the kernel centers $\boldsymbol{\mu}_X$ depending on the block of origin. As such, the modeling process is performed block-based but the model is still global. A pixel that lays on the border of a block thus receives influence of both submodels. In practice, however, it was found that many kernels then elongate along the block edges, which results in block artefacts in the reconstruction. This is partly mitigated by using overlapping blocks. The softmax weighing during the reconstruction then ensures a smoother transition from block to block.

As such, the problem is reduced from $O(NK)$ to $O(BN_bK_b)$ for B blocks N_b pixels and K_b kernels. For example, instead of modeling a 512x512 pixel image with $1,024$ kernels, we can now model using 64 blocks of 64x64 pixels with on average 16 kernels. Reducing the number of operations from 2^{28} to 2^{22} operations, or 2^{16} operations per block.

There are two major benefits to block-based modeling: (1) each block can be modeled in parallel which greatly reduces the total execution time, and (2) each block can receive a different kernel-budget. The latter was used in the first SMoE publication on grayscale 2-D images in which a 2D-DCT was performed on all blocks [54]. Based on the energy of the AC-coefficients, kernel-budgets were constructed to place more kernels into blocks with more spatial frequency energy, e.g. textured regions. The main disadvantages are that (1) kernels do not stretch beyond block borders, and (2) that blocks can still contain high number of samples when the coordinate space becomes higher-dimensional, e.g. light field video.

5.3.4 Minibatch EM

As discussed in Sec. 5.2.1, the standard EM algorithm (or batch EM) works as follows. In each iteration k, first the soft-membership $\hat{z}_{ij}$ of a pixel $i \leq N$ to each Gaussian $j \leq K$ is estimated, i.e. the likelihood of that sample originating from that Gaussian (E-step). Secondly, based on these soft-memberships the kernel

parameters are re-estimated based on the pixel data that belong to that kernel (or M-step):

$$\text{(M-step) } \Theta^{k+1} = \arg\max_{\Theta} \hat{z}_{ij}. \tag{5.8}$$

The earliest works on SMoE always relied on the batch version as described above [54], [56], [57]. However, in the application to light fields, the approach suffered from robustness issues when the amount of kernels became large [57]. The more kernels that are added, the more the optimization becomes sensitive towards local optima due to the vastly increasing number of parameters to optimize.

In order to mitigate these issues, a stochastic online version of the EM algorithm, or *minibatch* EM was proposed [60], [118], [119]. In minibatch EM, parameters are updated in a stochastic fashion by taking random minibatches of size m (randomly sampled pixels) and performing the M-step according to a learning speed η_k. A stepsize reduction power α was used as in the work by Liang and Klein to decrease the learning rate exponentially: $\eta_k = (k+2)^{-\alpha}$, with $0.5 < \alpha \leq 1$ [118]. The M-step is then calculated as follows.

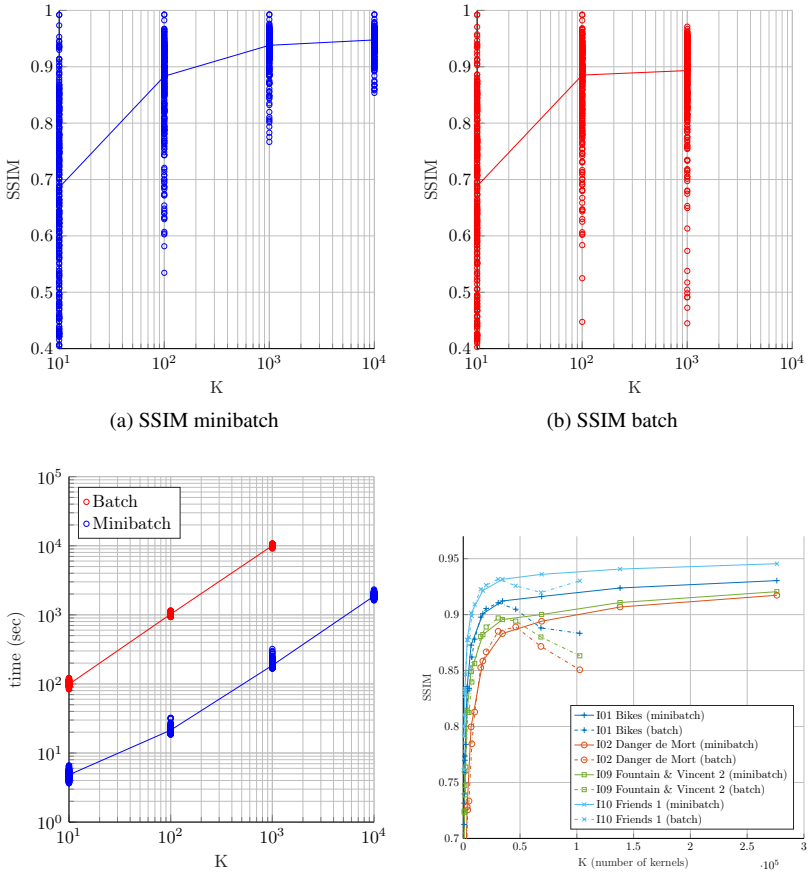
$$\Theta^{k+1} = (1-\eta)\Theta^{k} + \eta_k \left(\arg\max_{\Theta} \hat{z}_{ij}\right) \tag{5.9}$$

By using minibatches, the local optimum changes in every iteration as it is dependent on the seen data. As such, the assumption is that it converges more easily to a solution closer to the global optimum and thus behaves more robustly. Furthermore, as $m \ll N$, each iteration takes up N/m times less memory in the E-step and substantially lowers the duration of a single iteration. In the following, the above assumptions are validated using experiments.

5.3.4.1 Batch vs. Minibatch experiment

In this subsection, the minibatch modeling is compared to the batch modeling in terms of speed and reconstruction quality [57]. For these experiments, two datasets are used: (1) a new dataset with 324 small light field crops was extracted from the EPFL lenslet dataset used for ICIP Grand Challenge and the Call for Proposals for JPEG Pleno [131], and (2) five full LFs from the same EPFL dataset. The crops have 10-bit color depth, 64×64 image spatial resolution and an angular resolution of 13×13 camera coordinates. Each block thus contains 692,224 samples. The online EM introduces two new parameters: the batch size m and a driver for the learning rate α.

Fig. 5.1 shows the results for different model sizes $K = [10, 100, 1000, 10000]$. Consequently, given blocks of size $13 \times 13 \times 64 \times 64$, the sparsification ratio (pixels per kernel) is ranging from 1 kernel for $\pm$70,000 samples up to 1 kernel per 70 samples. From these results, it is apparent that for large K, the reconstruction quality becomes sensitive towards the values



(a) SSIM minibatch

(b) SSIM batch

(c) Time speedup 100×

(d) Light field modeling

Figure 5.2: Performance evaluation of batch EM vs. minibatch EM in terms of SSIM (a,b) and speed (c) on a dataset of light field crops. Note the logarithmic x-axis for (a), (b), and (c), as well as the logarithmic y-axis in (c). There are two things that are worth noting. First, when comparing the samples for batch (a) and the samples for minibatch (b), it is visible that higher quality is more consistently achieved for minibatch. In contrast to the batch method (a), where the quality could be unacceptable for even a large number of kernels for certain samples. Second, an impressive $\times 100$ speed-up is visible in (c). The reconstruction quality of complete light fields is validated in (5.2d). It is clear that the minibatch approach heavily increases the robustness for large Ks, while achieving a x100 speed-up.

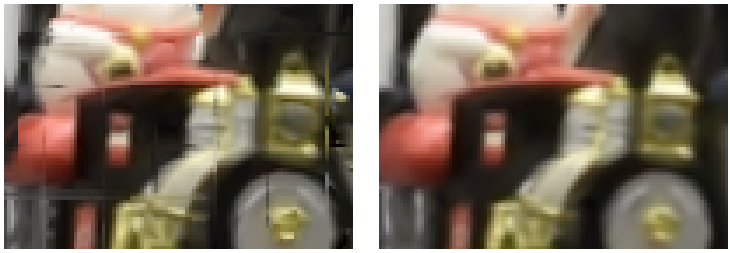
of (m, α), with differences of up to 10dB PSNR. A careful analysis of these parameters is thus advised. Empirically, the following parameters for blocks of $13 \times 13 \times 64 \times 64$ were found: $m = 1000$, and $\alpha = 0.5$ when $K < 1000$, and $\alpha = 0.8$ when $K > 1000$. Both the batch and the minibatch approaches are implemented using MATLAB.

Fig. 5.2 shows results of the reconstruction quality and modeling speed of the above mentioned dataset. These results show that the minibatch approach is up to $100\times$ faster for the same number of kernels K. Furthermore, the results confirm the increased robustness. The desired behavior of having monotonous positive relation between number of kernels K and reconstruction quality is experimentally confirmed. Given 10,000 kernels (1 kernel per 92 pixels), the minibatch EM algorithm reaches up to 37dB PSNR on average and 0.95 SSIM.

Using the local block-based modeling (with 3-pixel overlapping blocks) in Sec. 5.3.3, the performance of the modeling on the full LFs is compared. Firstly, it is clear that there is a strong increase of robustness. Whereas using the batch EM does not guarantee a monotonic increase of SSIM, the minibatch method does. Secondly, the strong decrease in runtime allows us to create models with a much higher number of kernels. To conclude, given careful a priori analysis of the hyperparameters, the usage of minibatches for training GMMs is beneficial in terms of speed, robustness and accuracy of reconstruction.

5.3.5 Split-and-Merge EM

In the first SMoE publication, an effective albeit computationally expensive modeling strategy was used in order to get out of local minima after convergence [54]. A split-and-merge approach was used to split undesired kernels, while merging others [132]. After EM convergence, two lists are maintained: (1) kernels that are likely to be split, (2) a list of pairs that could likely be merged. The splitting depends on a criterium, such as the conditional variance of the luma channel in $R_{Y|X}$ of that kernel. The conditional variance of the luma channel indicates the uncertainty of the prediction for that kernel. Splitting is performed along the most dominant axis with an offset [132]. The second list with possible merge-pairs is heuristically determined by evaluating the dot-product $\boldsymbol{z}_j^T \boldsymbol{z}_k$ for each kernel j and k ($j \neq k$), with $\boldsymbol{z}$ being the N-vector of responsibilities of that kernel for all pixels in that block. The idea is that the more overlap in responsibilities, the more likely the kernels should be merged [132]. The two lists are then combined to a list of split-and-merge candidates, i.e. 3-tuples: a kernel two split and two kernels to merge. The first split-and-merge candidate is tried, some EM iterations are performed and the reconstructed image is then compared. If the reconstruction quality is better, then proceed by establishing new lists. If not, then revert and choose another split-and-merge candidate.



(a) Block-based (b) Progressive

Figure 5.3: The hard block-level subdivision of the coordinate space results in visually disturbing artifacts when objects cross block boundaries in time. This is mitigated by the block-level updates that simulate global modeling.

The experiments had shown that this approach works well as it is sure to always improve the found solutions. However, it is computationally heavy and a very trial-and-test method. Nevertheless, this method is used in conjunction with the block-based method for coding of SMoE image models in Sec. 6.4. This method should be seen as more of a post-training refinement than a real modeling strategy.

5.3.6 Progressive modeling

The block-based method with overlapping blocks performs arguably well on static content, although with limited efficiency as kernels could not span multiple blocks and many kernels were spent on the block borders. However, as illustrated in Fig. 5.3, for modalities with a time dimension, the block-division results in disturbing block artifacts on moving objects. Therefore, in order to adequately model light field videos, a solution had to be found [58].

In this section, a computationally efficient and global EM variant is proposed for training SMoE models that also allows for an varying distribution of the kernels [58]. Firstly, global modeling is simulated by performing block-wise updates. Secondly, an iterative train-and-split strategy is implemented in order to achieve an varying distribution of the kernels. Table 5.1 summarizes the features of the discussed techniques.

5.3.6.1 Block-level updates

The following optimization allows us to drastically lower the computational demands for one minibatch iteration. The light field video is subdivided in overlapping spatio-temporal blocks, e.g. 32×32 pixels over 32 frames. These blocks are visited consecutively. A minibatch sample is selected from this block. The loglikelihood of each sample is determined by evaluating only the nearby relevant

	Global	Block-based	Progressive
Varying kernel density	low	high	high
Complexity	high	low	low
Block artifacts	no	yes	no

Table 5.1: Properties of SMoE modeling techniques: (1) global EM modeling on the entire image [56], (2) block-wise modeling which divides the image coordinate space into blocks and are trained independently [54], [57], and (3) the proposed progressive modeling that locally performs updates but simulates global modeling [55], [58]. Progressive modeling has the clear advantage of being low in complexity and enabling varying kernel density to match the local level of detail while simulating global modeling. The global modeling simulation resolves the block-artefacts that were present in the block-based method.

kernels. The loglikelihood of other kernels with these samples is considered to be zero. The relevant kernels are the kernels that have a center within a spatio-temporal relevance window. The kernels in the relevance window are updated after each block visit. As such, only the set of relevant kernels K_b and the m local minibatch samples are needed in memory. This results in $O(K_b m)$ per iteration per block, which heavily reduces the original requirements of $O(KN)$ per iteration. Note that kernels can migrate over the whole domain and can be present in several relevance windows in one image pass. The update factor η_k is divided by the number of blocks in the global image.

5.3.6.2 Kernel splitting

For the application of image approximation, it is desired to minimize the prediction variance of $Y|X = \boldsymbol{x}$. Previous research suggested the beneficial varying kernel spread property of initializing the EM algorithms using a split approach [133]. Borrowing ideas from these observations, a progressive modeling strategy was developed that progressively creates models with increasingly higher number of kernels where the prediction variance is high. This is done by splitting a certain amount of kernels based on the luma-channel of the weighted conditional variance $\pi_j R_{Y|X,j}$ (Eq. 3.11). Note that this calculation is significantly cheaper than the calculating the prediction error as in [133].

The modeling algorithm thus starts with an initial number of kernels K_{init} and model using block-level updates using minibatches. After convergence, the most uncertain kernels are split. These kernels are displaced from the original center along the splitting-dimensions and the covariances are scaled along the coordinate dimensions. These kernels then serve as a new initialization. This process is repeated until the number of kernels reaches a predetermined K_{max}. The splitting approach thus has the advantage that early in the process, less kernels are present. Consequently, the early iterations are considerably faster than later iterations as

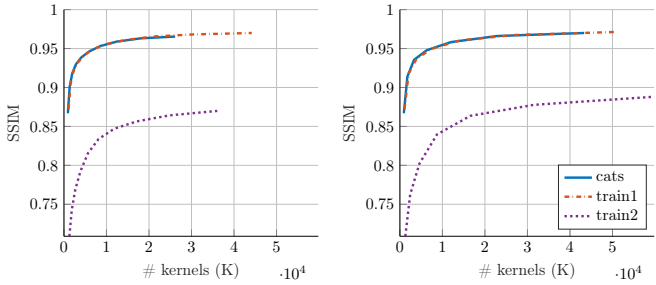


Figure 5.4: This figure illustrates the SSIM results for the three light field videos modeled using progressive modeling introduced in Sec. 5.3.6. The modeling was done by splitting respectively 10% (left) and 30% (right) of the top uncertain kernels.

K_b increases.

5.3.6.3 Example: modeling light field video

The exact splitting implementation depends on the image modality that is being modeled. In our work on light field video, we chose to split the kernels into 4 smaller kernels along the time and spatial dimensions (t, x_1, x_2) [58]. The reasoning is that in general, these dimensions have more chance to have smaller coherent segments compared to the camera dimensions (a_1, a_2), which typically have long-stretched kernels. The displacements for the four kernels are calculated based on the standard deviation along the coordinate dimensions, i.e. $\boldsymbol{\delta} = \sqrt{\text{diag}(R_{XX})}$ and the displacement was scaled per dimension, with $c_t = 0.05$, $c_{x_1} = c_{x_2} = 0.4$. The four new kernels $\boldsymbol{\mu}_{1,\dots,4}$ were displaced as follows:

$$
\begin{aligned}
\boldsymbol{\mu}^t_{X,1} &= \boldsymbol{\mu}^t_X - c_t\boldsymbol{\delta}^t, \quad \boldsymbol{\mu}^{x_1}_{X,1} = \boldsymbol{\mu}^{x_1}_X - c_{x_1}\boldsymbol{\delta}^{x_1}, \quad \boldsymbol{\mu}^{x_2}_{X,1} = \boldsymbol{\mu}^{x_2}_X + c_{x_2}\boldsymbol{\delta}^{x_2} \\
\boldsymbol{\mu}^t_{X,2} &= \boldsymbol{\mu}^t_X - c_t\boldsymbol{\delta}^t, \quad \boldsymbol{\mu}^{x_1}_{X,2} = \boldsymbol{\mu}^{x_1}_X + c_{x_1}\boldsymbol{\delta}^{x_1}, \quad \boldsymbol{\mu}^{x_2}_{X,2} = \boldsymbol{\mu}^{x_2}_X - c_{x_2}\boldsymbol{\delta}^{x_2} \\
\boldsymbol{\mu}^t_{X,3} &= \boldsymbol{\mu}^t_X + c_t\boldsymbol{\delta}^t, \quad \boldsymbol{\mu}^{x_1}_{X,3} = \boldsymbol{\mu}^{x_1}_X - c_{x_1}\boldsymbol{\delta}^{x_1}, \quad \boldsymbol{\mu}^{x_2}_{X,3} = \boldsymbol{\mu}^{x_2}_X - c_{x_2}\boldsymbol{\delta}^{x_2} \\
\boldsymbol{\mu}^t_{X,4} &= \boldsymbol{\mu}^t_X + c_t\boldsymbol{\delta}^t, \quad \boldsymbol{\mu}^{x_1}_{X,4} = \boldsymbol{\mu}^{x_1}_X + c_{x_1}\boldsymbol{\delta}^{x_1}, \quad \boldsymbol{\mu}^{x_2}_{X,4} = \boldsymbol{\mu}^{x_2}_X + c_{x_2}\boldsymbol{\delta}^{x_2}
\end{aligned}
$$

The covariance matrix for each new child kernel was scaled so that the sum of the child kernel "volumes" equal the total "volume" of the parent kernel. I put "volumes" in quotes as the Gaussian distribution has no geometrical concept of volume, however, the covariance matrices can be seen as definitions of the geometric ellipsoids. The rows and the columns of the (t, x_1, x_2) axes within R are scaled by $4^{1/3}$ as the scaling is to be performed along three axes and the kernels are split into four child kernels.

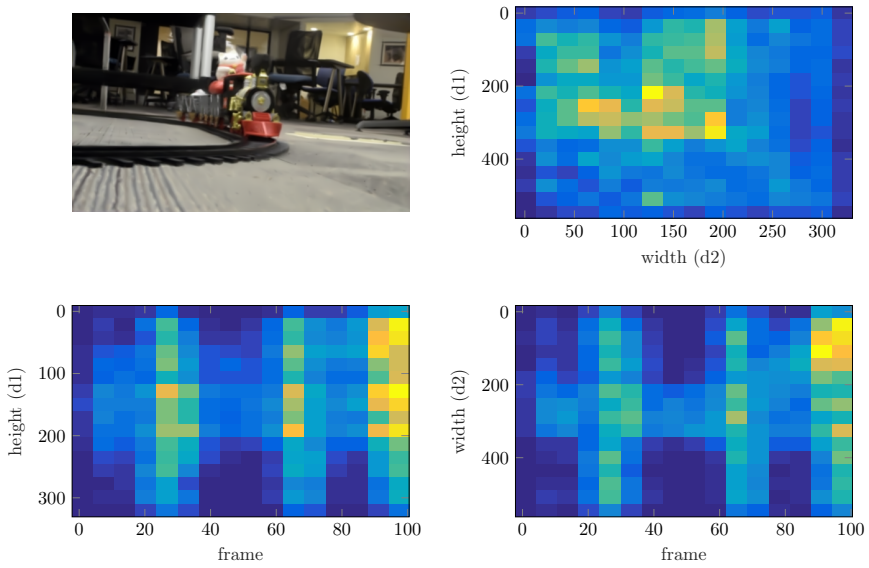


Figure 5.5: This figure illustrates the density of the kernels along the spatial and time dimension for a model trained on train2 *(top left) using $K = 49,455$ kernels. The density of kernels measures the number of kernels in a given image area. The light field depicts a toy train coming from the background center towards the front left. It is clear that the density is greater in areas with high motion due to the split operations. Spatially the kernels are concentrated top left where the train rides. Over time the train comes closer to the camera, this results in more kernels are being spent on the later frames. The train's trajectory is even visible on the bottom right density map. The train is first in the center and then moves towards the left (lower d_2 value).*

The three light field videos from Sec. 4.4 are selected: *cats*, *train1*, *train2* [115]. These videos all contain roughly 1 billion pixels. Fig. 5.4 shows the results for the three LF videos being modeled progressively. All models were initialized using $K_{\text{init}} = 2048$ and trained using $(s, \alpha) = (1\text{e}4, .6)$. Two splitting ratios were used: 10% and 30%. Due to the high number of views ($\pm 100 \times 8 \times 8$), the average SSIM was measured for a single view in each frame, rotating over the views $(2, 2), (3, 6), (4, 4), (7, 2)$.

The block-level updates were done using spatio-temporal blocks of $36 \times 36 \times 36$ pixels with an overlap of four pixels in each dimension. The kernel relevance window that determines which kernels to involve in this update was set to $54 \times 54 \times 54$ pixels. It is clear that subsequent models introduce a steady increase of reconstruction quality up to 0.97 SSIM. Fig. 5.4 also suggests that for this setup, the split-ratio is less important. As such, larger splits (right) do not seem to compromise the quality while requiring less iterations.

Fig. 5.5 illustrates the kernel distribution over the spatial and time dimensions

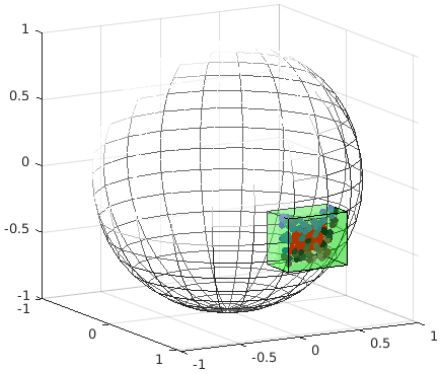


Figure 5.6: Illustration of the local updating on the sphere. In this step, the likelihood of the samples (red) are only being calculated by the kernels (colored) that lay in the vicinity of these samples. The relevance window is a cube (green) surrounding the samples.

for one model of *train2*. The vertical lines of high density of kernels along the time dimensions on the bottom row result from the block-level updates. In static areas kernels spread as far as possible in the time dimension as there is no change in color intensities. However, due to kernels falling outside of the relevance window, kernels in static regions still have a limited spread and other kernels take over. The position of the kernel centers on the angular dimensions are located near the middle, suggesting that each kernel has a maximal stretch along the angular dimensions, similar to Fig. 4.5. It can be concluded that each kernel specializes on the angular light information in one particular spatio-temporal region of varying size. Furthermore, the temporal variances stretch maximally as desired for kernels in static image regions.

5.3.7 Modeling spherical image dimensions

In Sec. 4.2, the extension of SMoE onto spherical dimensions was introduced. The basic idea is to operate in the 3-D Euclidean space in which the unit sphere is a manifold. The fact that all samples lay on that unit sphere has implications for the modeling process. A straightforward implementation of the above algorithms would result in many empty 3-D blocks and newly-split kernels should again lay on the unit sphere.

An iterative training method is proposed based on the covariance projection technique as described in Sec. 4.2 [55]. The proposed training method is further based on the progressive modeling approach as described above. In this approach, global modeling is simulated by a local fitting strategy in which a local group of samples is processed by only the set of relevant kernels in its vicinity. This simulates global modeling while minimizing the computational complexity. For

360-degree images, the kernel relevance window that selects the kernels becomes 3-D as shown in Fig. 5.6. Only a minibatch of these samples is selected in each iteration in order to increase the robustness of the kernel updates and to heavily decrease the computational complexity. This also allows us to uniformly sample the sphere while the input samples may have oversampled poles due to 2-D projections of the 360-degree image.

Models are initialized with K_{init} kernels spread out uniformly over the sphere and are further trained until convergence. After convergence, a portion of the top uncertain kernels are split into four smaller kernels which serves as a new initialization for the next meta-iteration. The split is performed on the tangent plane (Fig. 4.2) and the uncertainty is defined by the weighted conditional variance of the color space, i.e. $\pi_j \operatorname{Tr}(R_{Y|X,j})$. The new kernels are then projected back onto the unit sphere. This model is then further trained until convergence. The process stops when a predetermined K_{max} is exceeded (depending on the need). Finally, all the covariance matrices are projected onto the tangent planes and the centers are projected onto the unit sphere as detailed in Sec. 4.2. Note that the projection step does not introduce considerable computational overhead. The proposed method here was used to generate the results discussed in Sec. 4.2.4 and illustrated in Fig. 4.2.

5.4 Conclusion

It can thus be concluded that although the EM algorithm has unfavorable properties in terms of complexity, methods can still be devised for SMoE that can handle over billions of samples. The main idea is to carefully follow the structure of the input samples and perform local updates. This is made possible by the observation that a pixel is typically only dependent of a limited set of kernels in its vicinity and vice-versa. Furthermore, spatially (or temporal) neighboring pixels will likely have similar relevant kernels. Therefore, these pixels can be processed in batch with an identified set of neighboring kernels.

Nevertheless, the modeling is a time-consuming process. In this chapter, the focus did not lay on the exact timings or fast implementations, but rather on reducing the complexity from an algorithmic point of view. The algorithms in this dissertation were sequentially implemented using MATLAB on a single-thread CPU. The modeling could thus take up to several days. However, as mentioned, these algorithms have huge potential for massive parallelization on e.g. GPUs as each pixel is independently evaluated. For example, a pixel-parallel real-time rendering of SMoE models for large models was shown in which the same paradigm of local reconstruction was used [18].

6 SMoE for Compression and Coding

6.1 Introduction

This dissertation proposes the new unifying framework of SMoE for representing visual data. It was indicated that our model allows for a large number of applications, ranging from resampling, super-resolution, image description, depth estimation, segmentation, etc. However, the focus during the development of this dissertation has been coding of image modalities. Image and video coding has been around for decades. For example, the popular JPEG-format was standardized over 25 years ago [12]. T. Sikora's "Trends and Perspectives in Image and Video Coding" provides an historical overview of the evolution of image and video coding [16]:

> "Digital image and video coding research started in the 1950s and 1960s with spatial differential pulse-code modulation (DPCM) coding of images. In the 1970s, transform coding techniques were investigated. In 1974, Ahmed et al. introduced the famous block-based discrete cosine transform (DCT) strategy [134]. Motion compensated prediction error coding also started in the 1970s and matured into practical technology around 1985 with the advent of the basic hybrid block-based motion compensation/DCT systems (MC/DCT). MC/DCT coding strategies are implemented in all of today's MPEG and ITU video coding algorithms. [...]"

This remains true for even the latest video standards such as HEVC [17]. As

discussed in Chapter 1, the successor of the 3-D extension of HEVC (3D-HEVC) is the primary candidate for MPEG's 6-DoF vision [5], [8], whereas JPEG-Pleno is considering the 4D-DCT for light field coding [46]. Sikora ends his conclusion stating [16]:

> "While improved compression efficiency continues to be important for many applications, new functionalities and requirements will be imposed by user devices and network constraints. That is, emerging wireless image and video sensor networks requirements may drastically change existing coding paradigms."

In my opinion, this is exactly what is happening now, more than a decade after this publication. Video had a definite order of frames that are shown, however, 6-DoF VR has no such thing and therefore the efficiency of MC methods could be limited in the future. Furthermore, the sequential nature of the intra-prediction and MC limits the level of decoding parallelism to only the level of CPU-threads, whereas hardware now allows for massive parallelism. Furthermore, the approaches have very little applications in the compressed domain as only a low- to mid-level understanding is built of the data. The main exception being the motion vectors, which can be used for several post-processing tasks, e.g. video stabilization [135].

In this chapter, the coding application of the SMoE representation is detailed. A SMoE model consists of a set of kernels that is entirely defined by the kernel priors, their centers and their covariance matrices. The kernels were used to harvest pixel correlation over many dimensions simultaneously, however, the kernel parameters still exhibit redundancy. For example, nearby kernels are likely to have similar center coordinate and amplitudes. Furthermore, kernels that represent the same repeating structure are likely to have repeating spread in the coordinate dimensions. Note that when using SMoE for coding approaches that there are two phases where loss is introduced, i.e. the modeling and the lossy coding of the parameters. Coding an image modality using SMoE will thus always be a trade-off between the number of kernels and the quantization step sizes.

6.2 Relation to FTV and ray-space representation

During the development of this framework, it came to my attention that the proposed coding architecture in this chapter resembles an older proposal by Prof. Masayuki Tanimoto within MPEG-FTV, the free viewpoint television standardization efforts in MPEG. MPEG-FTV can be seen as the predecessor for later standards such as 3D-HEVC and the current MPEG-i Visual efforts. In FTV, everything revolved around the ray-space representation, a parametrization of the rays within a space as visualized in Fig. 6.1. The parametrization could be either

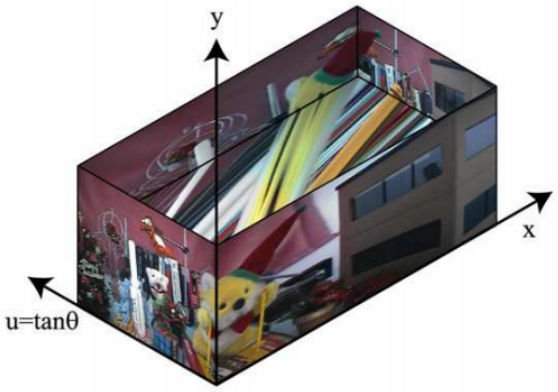
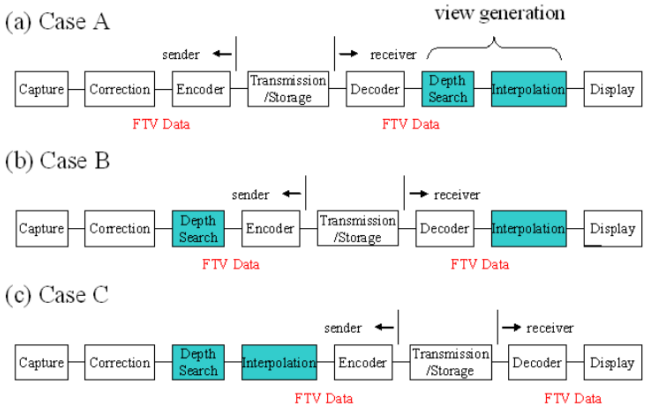


Figure 6.1: The ray-space representation, an alternative and very similar parametrization of the light rays in a space [136].

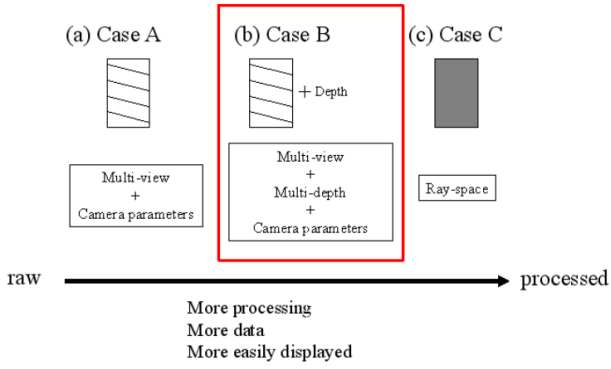
4-D (if the "open space" assumption is met) or defined as a 5-D space (similar to the proposal in this dissertation in Chapter 2), both defined for orthogonal and for spherical dimensions.

During the standardization for MPEG-FTV, three content-delivery architectures were proposed as illustrated in Fig. 6.2. The difference lies in the position of the depth-estimation and the ray-space interpolation steps. MPEG went for "Case B", in which the views are coded as multiview video with accompanying depth information [136]. This would then define the path of MPEG for the later efforts as well. A decision that makes much sense given the tools that were at hand and other efforts within MPEG at the time. Interestingly, "Case C" very much resembles the high-level idea behind the proposal in this dissertation. In Case C, a continuous representation of the light rays in a space is obtained at encoder side and is transmitted as such. Therefore, the decoder is spared of any view synthesis or interpolation processes. Additionally, this allows for using complex interpolation methods that are not real-time at encoding side. Fig. 6.2 from Tanimoto's proposal similarly indicates that "Case C" requires more processing, more data, but is more easily displayed.

In this chapter, a coding system is proposed that implements "Case C". The continuous SMoE model of the image modality is obtained at encoder side and the model parameters are further quantized and binarized to be transmitted over networks or to be stored. Note that in this chapter, the models were always built on densely sampled light fields which do not require any interpolation before SMoE modeling. Potentially, if the inter-camera distance becomes larger, it is possible that view interpolation would similarly be needed before SMoE modeling. Nevertheless, this dissertation does not cover sparse light fields.



(a) Possible architectures



(b) Corresponding data formats

Figure 6.2: This figure illustrates the three different content-delivery architectures and data formats that were proposed for MPEG-FTV, the predecessor for MPEG's immersive standards [136]. MPEG chose to go for option B, in which the interpolation of the ray-space representation is performed at the decoder side. Nevertheless, the proposed coding scheme in this chapter resembles option C, in which the ray-space is interpolated (being the SMoE model) and is transmitted as a continuous representation.

6.3 General Proposed Coding Structure

The nature of encoding a SMoE model is different compared to coding transform coefficients. In SMoE, the kernel parameters are coded and each parameter type has its own distribution, e.g. the priors have a very different distribution compared to for example, the luma amplitudes of all the centers. As such, the tendency is to encode the parameters per parameter type. Furthermore, each parameter type needs to be treated differently during quantization due to the distribution of that particular parameter group and depending on the sensitivity of our image reconstruction based on those quantized values.

6.3.1 Kernel centers

Let us first have a look at neighboring kernel correlation. First, kernel centers are typically locally correlated, i.e. kernels that lay in each others vicinity will have similar pixel coordinates and amplitudes. Secondly, the order in which the kernels are presented is currently not considered important. Therefore, the kernels can be sorted along an approximation of the shortest path that visits all kernel centers exactly once, starting with the kernel center that is closest to the origin. Finding such a path is equal to the traveling salesman problem, which is NP-complete in terms of complexity. Calculating the optimal solution is thus not achievable in an acceptable time. Heuristically, a greedy algorithm is employed that each time finds the closest kernel to the last kernel visited, which is computationally feasible.

The kernels are thus sorted by the centers $\boldsymbol{\mu} = [\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y]$ by defining a path that comprises every component once in a greedy fashion. Start with the component j closest to $(0, 0)$. Find component k $(k \neq j)$ so that $|\boldsymbol{\mu}_j - \boldsymbol{\mu}_k|$ is minimal. As such, each $\boldsymbol{\mu}_{j-1}$ is a good predictor for $\boldsymbol{\mu}_j$. Only the prediction error is then further quantized and binarized. Note that the prediction error $\boldsymbol{e}_j$ is calculated based on the dequantized $\tilde{\boldsymbol{\mu}}_{j-1}$ in order to prevent error propagation.

$$\boldsymbol{e}_j = \boldsymbol{\mu}_j - \tilde{\boldsymbol{\mu}}_{j-1} \tag{6.1}$$

This scheme is generally known as *Differential Pulse Code Modulation* (DPCM) and is illustrated by the feedback loop in the blockdiagrams Fig. 6.3 and Fig. 6.7. The prediction error vector $\boldsymbol{e}_j$ for each kernel j is thus $(p + q)$ dimensional. These error vectors typically follow a Laplacian distribution centered around zero.

6.3.2 Kernel covariance matrices

Non-neighboring kernel correlation is also present in a SMoE model as illustrated as follows. Remember, that kernels are steered along pixel correlation in every dimension. The coordinate covariance matrix R_{XX} defines the simultaneous spread

along all coordinate dimensions, or the "shape" of a kernel in the coordinate space. It is apparent that the probability of a specific kernel shape occurring in a model is not uniform. This is similar to traditional video coding where not all motion vectors in a video are equally probable as multiple blocks tend to follow similar motion. Similarly, when looking at the EPIs in light fields, it is equally clear that the distribution of the slopes is not uniform, but that there are certain classes of slopes depending on the depth of the corresponding object in the scene. Therefore, schemes can be devised to perform non-linear quantization by clustering of the possible kernel shapes.

As mentioned in Sec. 3.3.3, one of the advantages of using the mean estimator $E[Y|X = \boldsymbol{x}]$ is that it does not rely on the covariance color matrix R_{YY}. Even though this matrix enables other secondary functionality, e.g. indicating the prediction variance for each pixel, the R_{YY} matrix is not encoded in this chapter as it is unnecessary for the reconstruction and thus would result in unnecessary overhead.

The covariance matrices between coordinates and color amplitudes R_{XY} and R_{YX} are each others transposed, so only one needs to be transmitted. The values are used to calculate the slopes of the color gradients. These values are typically centered around zero (i.e. the constant regressor) following a Laplacian distribution. As experimentally shown in Sec. 3.5.4, there are different possible ways of representing and reconstructing the color channels. One particular interesting method that limits the number of necessary coefficients, is to not transmit the slopes, or only the slope for the luma gradients. In this chapter, experiments are performed to evaluate the RD performance of such methods.

6.3.3 Entropy coding

Entropy coding is at the core of virtually every image compression scheme. Entropy coding is a form of lossless coding that enables to write more common symbols using shorter bit-sequences. The length of the codeword is proportional to the negative logarithm of the probability of the corresponding symbol. In general, the Shannon entropy is defined using a discrete set of probabilities as follows:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2(p(x_i)). \tag{6.2}$$

The total entropy $H(X)$ tells us the theoretical minimum average symbol length in bits. The goal of an entropy coder is to approach this theoretical minimum and the goal is thus to transform our data in such a way that $H(X)$ becomes small. A typically desired distribution is one that has a strong peak for some symbols combined with very low probability for others, such as the Laplacian distribution. Luckily, as noted above, the outcome of the DPCM strategy, as well as the values

for R_{XY} follow such a Laplacian distribution. Both encoder and decoder need to know the approximate probability of each symbol. A good initial approximation enables the entropy coder to faster converge to the theoretical minimum.

In this work, an adaptive arithmetic coder is used that is initialized using a Laplacian distribution [137]. An adaptive encoder updates its internal distribution after each symbol to better fit the real distribution. Nevertheless, it converges faster to the entropy if the initial values distribution is close to the real distribution. Transmitting the probabilities for symbols can become problematic if the amount of symbols is high, e.g. an 8-bit image has 256 possible values. Therefore, if the distribution approximately follows a Laplacian, the scheme fits a Laplacian distribution on the real distribution. As such only the parameters (mean and variance) need to be transmitted to the decoder.

Due to these observations, the aim is to encode all these parameters in a single *adaptive arithmetic coder* (AAC) which assumes a Laplacian distribution. In order to do so, one option is to align all the distributions of the different types of parameters. Furthermore, it is desired to introduce more distortion in less important parameters in order to save bits. In order to align the distributions and to allow more distortion in some parameters than others, the following was proposed. The resulting parameters are first normalized to have zero mean and a certain standard deviation σ_i, with $\sigma_i \leq 1$. Then quantization is performed using a fixed quantization step $|min\_val, max\_val|/2^b$, based on the minimum and maximum value of all normalized parameters. The rationale is that parameters with $\sigma_i < 1$ are quantized with more distortion, in order to save bits on these less important parameters.

6.3.4 Coding summary

To summarize, the following choices can be made in order to save bits. Note that all choices lead to a possible visual degradation of the reconstructed image modality:

1. modeling using a lower number of kernels K;
2. decreasing the standard deviation σ_i of each parameter type i during normalization;
3. increasing the quantization step size by decreasing b.

6.4 Image Coding

In this section, the SMoE model is applied for coding 2-D color images. Fig. 6.3 depicts an overview of the proposed system. These results have not been published before, however, initial work on gray-scale 2-D images was published before [54]. In these experiments, the block-based EM modeling is used as described

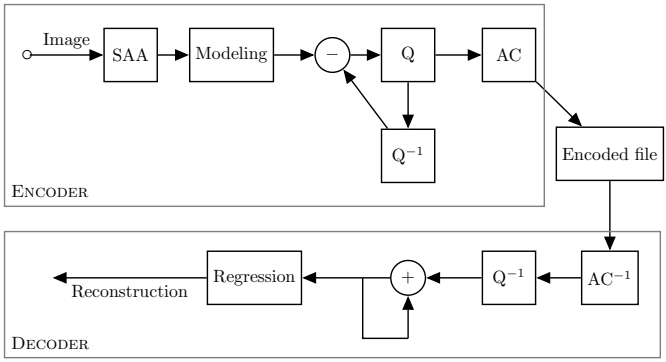


Figure 6.3: Overview of the coding process for 2-D images [54]. The encoder consists of an SAA *stage where a kernel budget is calculated for each block. The modeling is then performed block-based. Next, the model parameters are quantized and entropy coded. The decoder reconstructs the model parameters and then performs the regression to reconstruct the image.*

in Sec. 5.3.3. This modeling has the advantage that each block in the image can receive a different kernel budget. Therefore, first a *spatial activity analysis* (SAA) is performed to determine the budgets. The kernel centers are DPCM-coded. All parameters are further quantized and entropy coded. The decoder then performs the reverse actions and thus obtains the kernel parameters. These kernel parameters are then used to reconstruct the image through regression.

6.4.1 Spatial Activity Analysis and Modeling

The block-based EM modeling from Sec. 5.3.3 algorithm is used to estimate the parameters $\Phi_j = (\pi_j, \mu_j, R_j)$ for every kernel j. In order to avoid local optima, a split-and-merge approach from Sec. 5.3.5 was used to split undesired components, while merging others [132]. The modeling task is subdivided into overlapping blocks. This is in contrast to the non-overlapping blocks in our first SMoE-related work [54]. The overlap mitigates the abrupt changes around block-edges previously visible, which are due to data truncation. Kernels that lay inside of the overlap are discarded as they are assumed to be present in the adjacent block. Consequently, each block can see into its neighboring pixels, but can not place any components there. In general, it is important to arrive at few components in regions that are flat, but a larger amount in detailed areas. Every block receives a different budget of components in order to achieve this. Similar to [72], a 2D-DCT is performed and the spatial activity A_i for block i is calculated as the normalized squared sum of the first row and column of the AC coefficients. As such, flat areas receive less kernels, while more highly textured regions receive more. Given

the average kernels per block K, and a spatial activity sensitivity parameter τ, the budget K_i for every block i is calculated as

$$K_i = K + \text{round}(\tau(A_i - E[A])K) \tag{6.3}$$

Note that the modeling is performed block-wise, but the reconstruction is global.

6.4.2 Difference coding and quantization

The centers $\boldsymbol{\mu} = [\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y]$ are difference coded by defining a path that comprises every component in a greedy fashion. Start with the component j closest to $(0, 0)$. Find component $k, (k \neq j)$, so that $|\boldsymbol{\mu}_j - \boldsymbol{\mu}_k|$ is minimal.

At the decoder side, only $R_{XX,j}$ and R_{XY} are needed for reconstruction of the images. Quantizing the covariance coefficients directly can easily lead to non-positive definite matrices, which are required. An Eigen-decomposition allows for a more robust quantization. Accordingly $R_{XX,j}$ is coded as α_j, the angle of the eigenvector placed in the first quadrant, combined with $v_{j,1}, v_{j,2}$, the corresponding eigenvalues. α_j and $v_{j,1}, v_{j,2}$ are uniformly quantized.

6.4.3 Entropy coding

The 5-D DPCM prediction errors $\boldsymbol{e}_j = [e_{1,j}, ..., e_{5,j}]$'s (2 coordinates, 3 color channels) are assumed to be Laplacian distributed and are normalized to obtain a standard deviation of σ_i for the ith component in $\boldsymbol{e}_j$, with $c_i \leq 1$ being the ratio determining how much more subsampled the coefficient i needs to be compared to the pixel coordinate. In this work, a baseline is set as $\sigma_1 = \sigma_2 = 1$, i.e. σ_i with $i > 2$ determines how much less important coefficient i is compared to the location centers. This assumes that the precision of the location will always be the highest compared to the other coefficients. Consequently, the distribution of the coefficient i with $\sigma_i < 1$ is squeezed together. Next, quantization is performed uniformly based on the limits of location centers $\boldsymbol{\mu}_{X,j}$. As such, it is possible to combine different quantization step sizes for each coefficient, while still using a single arithmetic coder. For this, the same Laplacian adaptive arithmetic coder is employed as in [72]. Analogously, the same was done for 2-D covariance vectors between coordinates and each color channel YCbCr: $R_{X\text{Y}}$ compared to $R_{X\text{Cb}}$ and $R_{X\text{Cr}}$.

In contrary to previous work [54], the priors π_j are not estimated at decoder side. Instead, the models were trained by constraining the priors to be $1/K$. Note that both the modeling and the coefficient quantization contribute to the reconstruction error.

6.4.4 Image coding experiments

The above coding approach is evaluated on the standard 512-by-512 color images *Lena*, *Baboon*, and *Peppers*. Based on the observations in Sec. 3.5.4, three coding modes are evaluated: (1) coding of full 2x3 covariance matrices R_{XY}, (2) constant color regressors, and (3) constant regressors (no covariance). As such, less information needs to be stored for modes (2) and (3) as the covariances between coordinates and chroma (or coordinates and luma) can be set to zero.

Block sizes for SAA were in the range [32,64,128]; τ varied between 1.5 and 3; and between 8 and 48 components per block were used for coding and reconstruction. A maximum of 12 split-and-merge operations and 150 EM-iterations were performed. Optimization was done using $\text{SSIM}_{\text{YCbCr}} = (2\ \text{SSIM}_{\text{Y}} + \text{SSIM}_{\text{Cb}} + \text{SSIM}_{\text{Cr}})/4$.

RD-results are shown in Fig. 6.4 and indicate that the SMoE coding approach can substantially outperform JPEG at low rates, both in terms of PSNR and SSIM. Gains are achieved for rates below 0.3 bpp. It is my opinion that this is a promising result, given that this pixel-domain coding approach departs so drastically from the existing frequency-domain JPEG and JPEG-2000 coding methods developed over the last 30 years. The non-optimized coding approach could at this early stage of implementation however not compete with JPEG-2000. On the *Baboon* image, the results were less impressive. This image is highly texturized and requires a large number of steering kernels to arrive at a sufficient quality. For the *Lena* and *Peppers* images, a sparse representation is more adequate as the required spatial bandwidth varies over the image.

It appears that RD-curves rise slower compared to JPEG and JPEG-2000 for higher rates. The SMoE models become too elaborate to be coded efficiently. For textured regions, many components are needed to exactly replicate the texture data, which become excessively expensive. Interestingly, the results show that for this dataset, it is beneficial to set the color gradients to constant, but to keep the luminance gradients.

Fig. 6.5 shows a side-by-side comparison between JPEG and SMoE. First, the decoded *Lena* images are shown around 0.15-0.17 bpp. At this rate JPEG results in strong block artifacts while SMoE results in a smooth representation. SMoE does capture all dominant structures very well. Second, the fine texture in *Baboon* is hard to approximate by a piecewise linear regression and yields modeling compression artifacts. JPEG thus heavily outperforms SMoE for the *Baboon* image. Finally, the *Peppers* image is shown at at 0.45 bpp. The objective quality metrics yield slightly higher results for JPEG. However, SMoE results in a very smooth reconstruction, whereas JPEG still shows block artifacts and artifacts around edges.

A comparison between coding modes (1) and (3) is visualized in Fig. 6.6. The full covariance mode (1) in Fig. 6.6 (left) results in smooth transitions within regions using the piecewise-linear smooth regression, while the mode (3) (right)

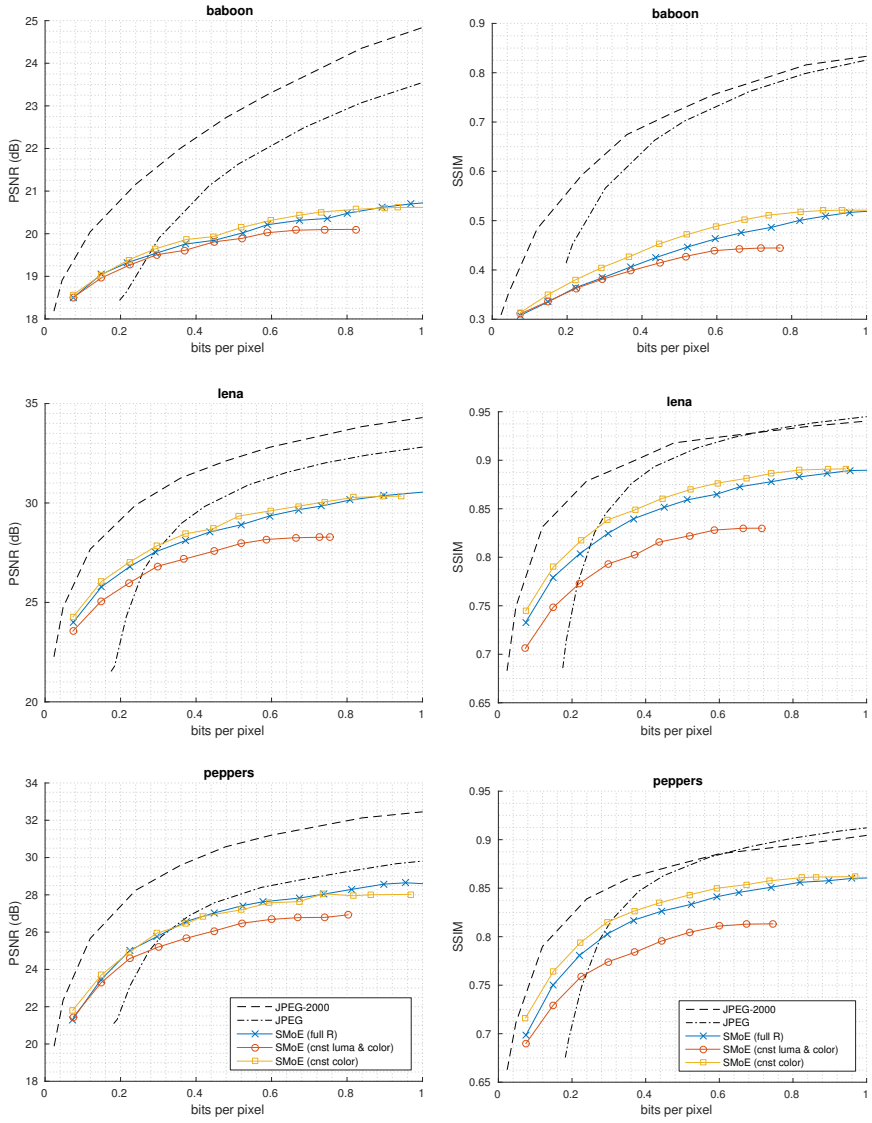
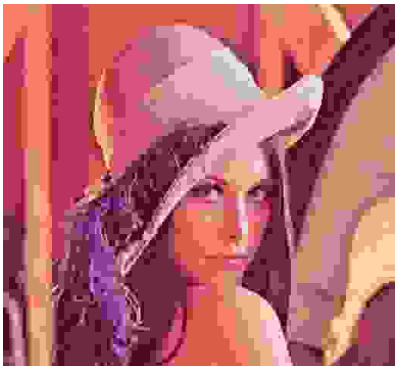


Figure 6.4: Rate-distortion curves for Lena, Baboon, *and* Peppers *in terms of PSNR (left) and SSIM (right). In general, SMoE outperforms JPEG for low bitrates, but SMoE is consistently outperformed by JPEG-2000. The* Baboon *image mainly contains high-frequenced textures that are not well captured by the SMoE model which results in poor RD-performance.*



(a) Lena (JPEG) at 0.17 bpp: 21.53 dB (PSNR), 0.68 (SSIM)

(b) Lena (SMoE) at 0.15 bpp: 26.05 dB (PSNR), 0.79 (SSIM)

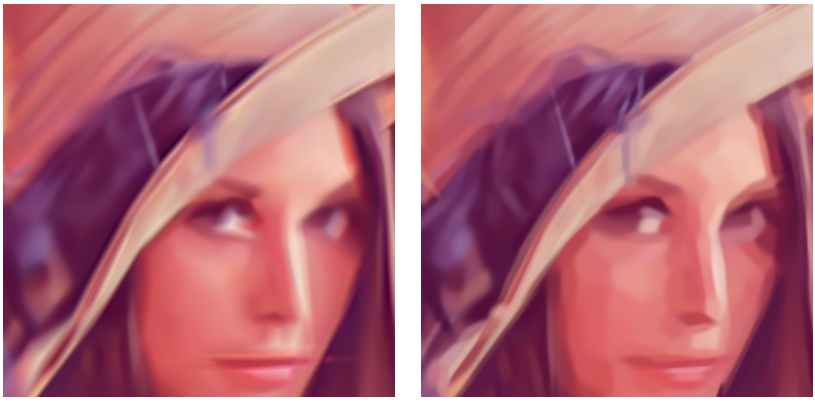
(c) Baboon (JPEG) at 0.43 bpp: 21.12 dB (PSNR), 0.66 (SSIM)

(d) Baboon (SMoE) at 0.45 bpp: 19.85 dB (PSNR), 0.42 (SSIM)

(e) Peppers (JPEG) at 0.45 bpp: 27.57 dB (PSNR), 0.86 (SSIM)

(f) Peppers (SMoE) at 0.45 bpp: 27.03 dB (PSNR), 0.82 (SSIM)

Figure 6.5: Visual comparison between JPEG and SMoE. Note how JPEG typically results in block artifacts at low bit rates, whereas SMoE provides a edge-aware smoothed reconstruction.



(a) SMoE with full covariance at 0.224 bpp: 26.79 dB (PSNR), 0.80 (SSIM)

(b) SMoE with constant regressors at 0.223 bpp: 25.98 dB (PSNR), 0.77 (SSIM)

Figure 6.6: Visual comparison between (left) full covariance (i.e. gradients for all three color channels per kernel) and (right) constant regressors for each color channel, including the luma-channel. In this example, the kernel boundaries are visible when using constant regressors, whereas those borders are not visible when incorporating gradients. Losing the gradient information in the chroma planes is less visually deterring compared to losing all gradient information.

results in piecewise-constant reconstruction of regions similar to the "cartoon-like" reconstruction with the Mumford-Shah-based image segmentation approach. SMoE yields a soft-segmentation, labeling each segment with the color of its center.

6.5 Light Field Coding

In this section, the SMoE coding approach tailored to light field models is presented. Due to the higher coordinate dimension, some parameters are handled differently. More specifically, the coordinate covariance matrix R_{XX} is now a 4×4 matrix. The Eigen-decomposition method that was used for 2-D images is replaced as R_{XX} becomes hard to describe in terms of rotations in four dimensions. It is replaced by a dimension-agnostic non-linear quantization method described in the following subsection. Fig. 6.7 shows an overview of the encoding process for light fields.

6.5.1 Window $R_{XX,j}$ quantization

As shown in Fig. 6.8, there is a high level of redundancy in the shapes of the kernels which are defined by the covariance in the coordinate space, i.e. $R_{XX,j}$.

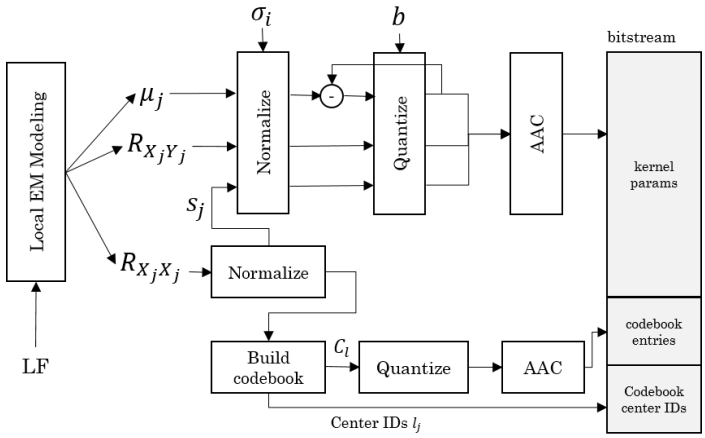
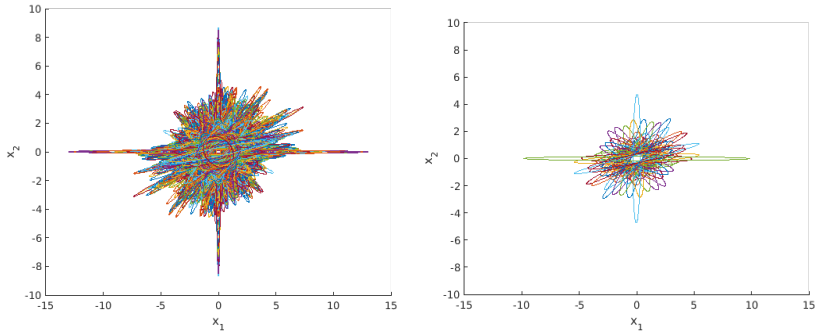


Figure 6.7: The block diagram of the proposed encoding process. The 4-D LF is modeled in a blockwise manner and results in the parameters $\theta_j = \{\boldsymbol{\mu}_j, R_{XX,j}, R_{XY,j}\}_{j=1}^{K}$ for all K kernels (other parameters are not coded). Firstly, all except $R_{XX,j}$ are coded similarly, each parameter type i is normalized to have zero mean and a standard deviation σ_i (according to the importance of that parameter). These parameters are then quantized and coded in a single bitstream using an adaptive arithmetic coder (AAC). Secondly, $R_{XX,j}$ is first scaled by $1/s_j$, with $s_j = |R_{XX,j}|^{1/4}$ so that each determinant equals one. The normalized $R_{XX,j}$ are then used to build a codebook with centers C_l. The codebook centers C_l are then quantized and arithmetic encoded. Finally, for each kernel j, the index l_j is encoded of the nearest cluster C_l.



(a) All 8960 R_{XX} covariances (b) Generated 64-entries dictionary

Figure 6.8: On the left side, all normalized $R_{XX,j}$ of a single light field model ($K = 8960$) are illustrated as ellipses. The $R_{XX,j}$ is the covariance in coordinate dimensions and defines the spread of each kernel in the coordinate space. For illustration purposes, only the two spatial dimensions (x_1, x_2) are shown. It is clear that there is a high level of redundancy. In order to reduce the redundancy in possible kernel shapes, a codebook algorithm was developed. The right plot illustrates the resulting codebook with only 64 dictionary entries. Codebooks are trained and binarized per model.

Therefore, a vector quantization-like method is employed for coding the window covariance $R_{XX,j}$. An EM-like algorithm is proposed based on the *Kullback-Leibler* (KL) divergence. As such, the probability densities are compared, which are more informative than the covariance parameters. Thus, instead of coding the 4x4 matrix $R_{XX,j}$, three items need to be encoded: (1) the smaller codebook with L entries $C_l, l \leq L$, (2) the index l of the closest cluster center for each kernel j and (3) a scale s_j for each kernel, as the codebook entries are normalized. The assumption is that similar reconstruction quality can be achieved with $L \ll K$.

All $R_{XX,j}$ are normalized by $|R_{XX,j}|^{1/d}$. In the case of $R_{XX,j}$ for 4-D LFs, d equals 4. As such, the constructed codebook contains normalized shapes with a determinant of one. The coding of the magnitude of the shape, i.e. $s_j = |R_{XX,j}|^{1/d}$ is discussed in the next subsection.

The KL-divergence for multivariate Gaussians $A \sim \mathcal{N}(\boldsymbol{\mu}_A, R_A)$ and $B \sim \mathcal{N}(\boldsymbol{\mu}_B, R_B)$ is given by

$$\mathrm{D}_{\mathrm{KL}}(A \parallel B) = \frac{1}{2}\left[\log\left(\frac{|R_A|}{|R_B|}\right) - d + \mathrm{tr}(R_B^{-1}R_A)\right] + \frac{1}{2}\left[(\boldsymbol{\mu}_B - \boldsymbol{\mu}_A)^T R_B^{-1}(\boldsymbol{\mu}_B - \boldsymbol{\mu}_A)\right] \quad (6.4)$$

As the covariance matrices are normalized a priori, $|R_A|$ and $|R_B|$ equal one. Furthermore, the windows are assumed to be centered on the origin, i.e. $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ are zero. In order to obtain a symmetric similarity measure, the distance metric

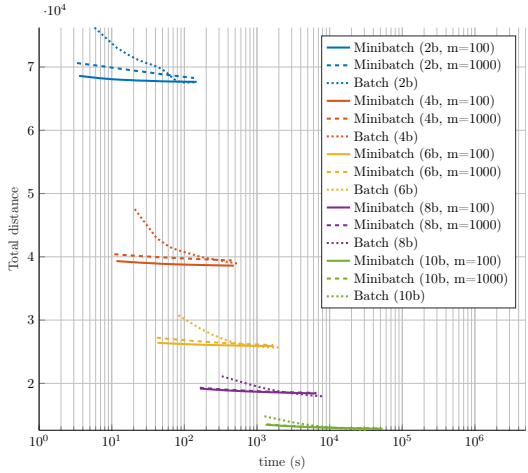


Figure 6.9: A k-means algorithm based on the Kullback-Leibler divergence is used in order to train codebooks as in Fig. 6.8. However, the batch method is slow to converge (dotted line) [57]. Similar to the used minibatch EM, a minibatch version was developed for this codebook training using per-cluster learning rate [138], which converges much faster than the batched version as illustrated here. Codebooks have 2^b entries. The figure also shows that, when b increases, the total distance lowers, but it also takes longer to train a codebook. Note that the x-axis is in log-scale. The y-axis indicates the total distance of all $R_{XX,j}$ to their corresponding codebook entry.

is defined as

$$\mathrm{d}(A,B) = \frac{\mathrm{D_{KL}}(A \parallel B) + \mathrm{D_{KL}}(A \parallel B)}{2} \tag{6.5}$$

$$= \frac{1}{4}\left(-2d + \mathrm{tr}(R_B^{-1}R_A) + \mathrm{tr}(R_A^{-1}R_B)\right) \tag{6.6}$$

Covariances are clustered around a centroid using $\mathrm{d}(A,B)$. At each iteration, the new centroid covariance C_l is calculated as the mean covariance of the members of the cluster l, and renormalized. Note however, that the mean is not optimal in terms of symmetrized KL-divergence as can be shown as follows. Note that Eq. 6.6 simplifies to the following equation in 1-D.

$$d(A,B) = \frac{1}{4}(-2 + \frac{\sigma_A}{\sigma_B} + \frac{\sigma_B}{\sigma_A}) \tag{6.7}$$

Consider two Gaussians with zero mean and standard deviations σ_A and σ_B. Take centroid with standard deviation

$$\sigma_C = \frac{\sigma_A + \sigma_B}{2} \tag{6.8}$$

If the new centroid would be optimal in terms of symmetrized KL-divergence, then the distance between A and C should be equal to the distance between B and

	bpp	K	L	b	σ_A	σ_Y	σ_{UV}	σ_s	σ_{XY}	σ_{AY}	PSNR (dB)	SSIM	MOS	CI
I01	0.006	1393	1024	12	0.50	1.00	0.12	0.12	0.50	1.00	27.37	0.756	1.90	0.149
	0.030	17188	1024	10	0.50	0.50	0.12	0.50	0.50	0.25	31.24	0.873	3.77	0.181
	0.098	30755	4096	14	0.50	0.25	0.25	0.50	0.12	0.50	32.68	0.898	4.00	0.164
	0.276	118846	4096	12	0.50	0.25	0.12	0.12	0.50	1.00	33.21	0.906	4.16	0.143
I02	0.006	1386	1024	12	0.50	0.50	0.25	1.00	1.00	0.50	25.55	0.629	1.52	0.138
	0.028	17202	64	10	0.25	0.50	0.25	0.25	0.12	0.50	28.90	0.795	2.97	0.223
	0.092	31946	2048	14	0.25	1.00	0.12	0.12	0.25	0.12	31.16	0.864	4.03	0.105
	0.346	121649	4096	14	0.25	0.50	0.12	0.25	0.12	0.50	32.16	0.884	4.29	0.145
I03	0.006	3455	64	10	0.25	0.25	0.12	0.25	0.12	0.50	26.00	0.643	1.90	0.196
	0.030	17208	64	10	0.25	1.00	0.12	0.25	1.00	0.25	28.48	0.790	2.84	0.190
	0.076	32146	4096	12	0.50	0.25	0.12	0.50	0.25	0.12	30.66	0.863	3.94	0.233
	0.280	121720	2048	12	0.25	1.00	0.25	1.00	1.00	0.25	31.56	0.885	4.13	0.205
I04	0.006	1381	1024	14	0.50	1.00	0.12	1.00	0.12	0.50	29.13	0.717	2.03	0.246
	0.030	17204	1024	10	0.50	0.50	0.12	0.12	0.50	0.50	32.00	0.831	3.77	0.181
	0.092	30563	4096	14	0.25	0.25	0.12	0.25	0.25	0.50	33.21	0.865	4.16	0.120
	0.358	118239	4096	14	0.50	0.50	0.12	0.50	1.00	1.00	33.82	0.879	4.26	0.117
I10	0.006	3373	64	10	0.25	0.25	0.12	0.50	0.50	0.25	30.59	0.850	1.90	0.173
	0.028	16913	1024	10	0.25	1.00	0.12	0.12	0.50	0.50	33.51	0.907	3.74	0.210
	0.089	30724	4096	14	0.25	1.00	0.12	0.12	1.00	0.12	34.81	0.926	4.13	0.135
	0.309	117135	4096	14	0.50	1.00	0.12	0.50	0.50	0.50	35.39	0.933	4.32	0.126

Table 6.1: Coding parameters, objective (PSNR, SSIM) and subjective (MOS, confidence interval CI) quality results for SMoE. The models contain K kernels. The covariance matrix R_{XX} is normalized by dividing the scale $s = |R_{XX}|^{1/4}$. The normalized R_{XX} is then coded using a dictionary with L entries. The identifier of the closest kernel in that dictionary is then coded using $\log_2(L)$ bits. The dictionary is trained on the set of normalized R_{XX} of this specific model and is transmitted along with the other parameters. All other parameters are normalized to have zero mean and a certain standard deviation σ depending on the parameter type. The next step is to quantize using a fixed quantization step by dividing the maximum range into 2^b steps. All σ are ≤ 1. The rationale is that parameters with $\sigma < 1$ are quantized with more distortion, in order to save bits on these less important parameters. σ is 1, except for σ_A, σ_Y, σ_{UV}, being respectively the 2-D camera coordinates part of the difference coded μ_X, μ_{Y_Y}, and $(\mu_{Y_{Cb}}, \mu_{Y_{Cr}})$. Next, σ_s, σ_{XY}, σ_{AY} are the σ-values for respectively the scale s per kernel, the covariance between spatial dimensions and luma channel, and the covariance between angular and luma channel. All parameters are then arithmetic coded, assuming a Laplace distribution as initialization.

C. Basically, $\frac{2\sigma_B}{\sigma_A+\sigma_B} + \frac{\sigma_A+\sigma_B}{2\sigma_B}$ should be equal to $\frac{2\sigma_A}{\sigma_A+\sigma_B} + \frac{\sigma_A+\sigma_B}{2\sigma_A}$. Take σ_A equal to 1 and σ_B equal to 2. Then we arrive at $\frac{25}{12}$ and $\frac{26}{12}$. The distances are thus not equal and C is thus not in the middle in terms of symmetrized KL divergence. However, in this case, the new centroid lays in between the two original Gaussians and relatively close to the middle.

This codebook was trained at encoder side, and transformed to ensure robustness. As each C_l is semi-positive definite, C_l can be decomposed using Cholesky: $C_l = U^T U$. U is vectorized and each coefficient is coded analogously to the slopes $R_{XY,j}$ (see Sec. 6.5.2). At decoder side, the multiplication $U^T U$ ensures the reconstructed covariance to be semi-positive definite again.

At decoder side, $R_{XX,j}$ is thus reconstructed as follows:

$$\tilde{R}_{XX,j} = s_j \times C_{l_j}, \tag{6.9}$$

with l_j the index of the closest codebook center C to the original normalized kernel covariance matrix and is coded using $\log_2(L)$ bits per kernel.

However, the outlined algorithm is known to scale badly to high numbers of kernels. At each iteration, the distance is between each kernel's $R_{XX,j}$ and codebook center's C_l is calculated (cfr. Sec. 5.3.4.1). Similar to the minibatch approach for the EM algorithm, a minibatch codebook training method was developed by employing a per-cluster learning rate as in [138]. Fig. 6.9 illustrates the convergence of the batch and minibatch versions of the algorithm. The total distance of all covariances compared to the dictionary is used as the metric. The dictionaries are of sizes $[2^2, ..., 2^{10}]$ trained on a set of 100.000 covariances comparing batch k-means, and minibatch k-means with minibatch sizes $m = [100, 1000]$. It is clear that larger dictionaries result in drastically lower total distances. Furthermore, it shows that the minibatch approach converges much faster than the batch approach while requiring vastly less memory.

6.5.2 Center and slope quantization and arithmetic coding

Identical to the 2-D image case, DPCM is performed on the sorted kernel centers. The kernels are sorted by the centers $\boldsymbol{\mu} = [\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y]$ by defining a path that comprises every component once in a greedy fashion. Start with the component j closest to $(0, 0)$. Find component k ($k \neq j$) so that $|\boldsymbol{\mu}_j - \boldsymbol{\mu}_k|$ is minimal. As such, each $\boldsymbol{\mu}_{j-1}$ is a good predictor for $\boldsymbol{\mu}_j$. Only the prediction error is further quantized and binarized. Note that the prediction error $\boldsymbol{e}_j$ is calculated based on the dequantized $\tilde{\boldsymbol{\mu}}_{j-1}$ in order to prevent error propagation.

$$\boldsymbol{e}_j = \boldsymbol{\mu}_j - \tilde{\boldsymbol{\mu}}_{j-1} \tag{6.10}$$

This DPCM approach is illustrated by the feedback loop in Fig. 6.7. The resulting prediction error vector is consequently 7-D for each kernel j. These vectors typically follow a Laplacian distribution.

Secondly, the full 4x3 covariance matrix $R_{XY,j}$ is not entirely encoded. As this experiment operates in the 3-D YCbCr space, the goal is to only encode the gradients along the luma channel. The scheme thus continues working with a 4x1 covariance matrix, the other elements are assumed to be zero. From our tests, it was observed that the remaining values naturally follow a Laplacian distribution. The final parameter to be encoded is the magnitude of the covariance matrix $R_{XX,j}$, which is $s_j = |R_{XX,j}|^{1/4}$. This parameter naturally follows a distribution close to a positive-Laplacian distribution.

A total of 12 parameters are thus encoded per kernel j, i.e. the 7-D prediction errors $\boldsymbol{e}_j$, the 4 dimensions in $R_{XY,j}$ (discarding the chroma dimensions) and the shape magnitude s_j. Due to these observations, we aim to encode all these parameters in a single *Adaptive Arithmetic Coder* (AAC) which assumes a Laplacian distribution. Therefore, we need to align all the distributions of the remaining 12 parameters. Furthermore, we want more distortion in less important parameters in order to save bits.

In order to align the distributions and to allow more distortion in some parameters than others, we propose the following. The resulting parameters are normalized to have zero mean and a certain standard deviation σ_i, with $\sigma_i \leq 1$. A fixed quantization step $|min\_val, max\_val|/2^b$ is used based on the minimum and maximum value of all normalized parameters. The rationale is that parameters with $\sigma_i < 1$ are quantized with more distortion, in order to save bits on these less important parameters. Finally, the quantized values are entropy coded by employing an adaptive arithmetic coder which is initialized by a Laplacian distribution.

6.5.3 Light field coding experiments

In Sec. 5.3.4, the minibatch EM approach was evaluated on a dataset of light fields. Furthermore, Fig. 5.2 showed the superiority in robustness of using minibatch EM vs. full-batch EM when using the block-based modeling strategy. In this section, a coding method for SMoE LF models is presented based on the models resulting from the batch vs. minibatch experiments in Sec. 5.3.4.

The following parameters were found using random search: block size , kernels per block K_i, quantization steps, and codebook size. The block size for the minibatch was fixed to 64 with K_i between 10 and 4000, whereas the block size for the batch EM ranged $[11, 17, 21, 32, 64, 128]$ with K_i between 6 and 48. The quantization step ranged $[10, 12, 14]$, ratios $\sigma_i = [1, 1/2, 1/4, 1/8]$, and book sizes $L = [2^6, 2^8, 2^{10}, 2^{11}, 2^{12}, 2^{13}]$.

The largest portion of the computational complexity is situated in the local modeling and the codebook building and is very dependent on the number of kernels per block in modeling (see Fig. 5.2c). A model of 30K kernels requires two hours, whereas 270K kernels requires three days. The training of the codebooks depends on the number of kernels K and size of the codebook L, and range between some minutes and two hours. Note that this is non-optimized code in MATLAB running on a single thread of a Intel™Xeon™CPU E5-2650 v3 @ 2.30GHz machine. Reconstruction can be done in real-time [18].

For comparison, the light fields were encoded as HEVC videos using the reference encoder HM-16.17 [139]. In order to have a logical ordering, the video is built by traversing in a snakelike-manner from the top left view towards the bottom right view. In order to ensure a fair comparison, the outer most views were not en-

(a) I01 Bikes

(b) I02 Danger de Mort

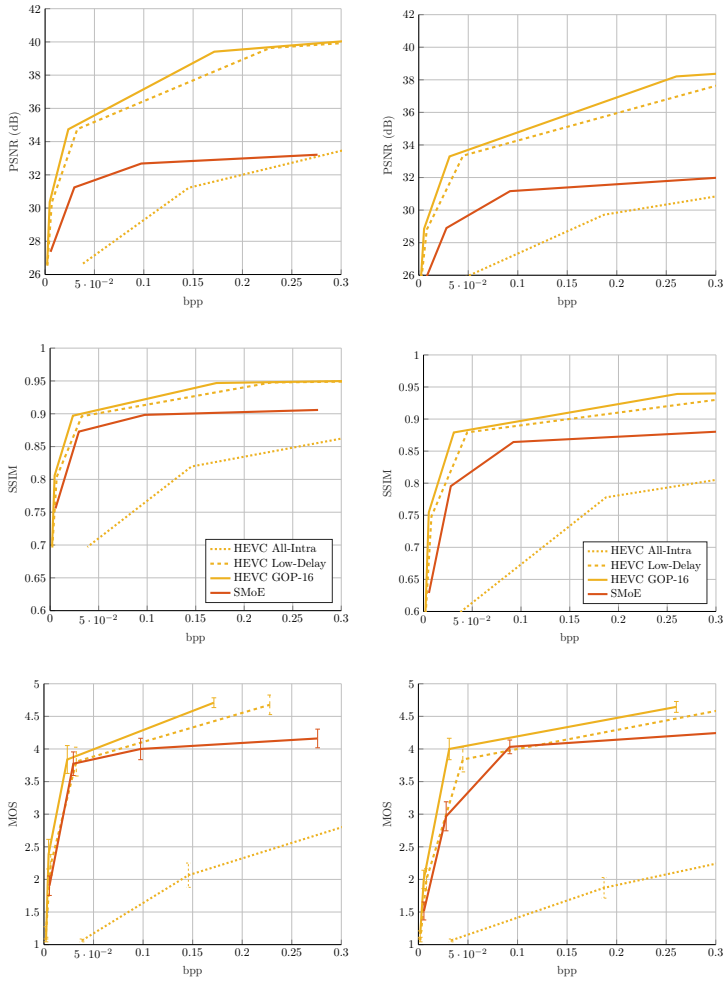


Figure 6.10: Coding results comparing the three HEVC (All-Intra, Low-Delay, and GOP-16) and SMoE in terms of PSNR, SSIM, and MOS scores for the light fields I01 Bikes *and* I02 Danger de Mort.

(c) I03 Flowers

(d) I04 Stone Pillars Outside

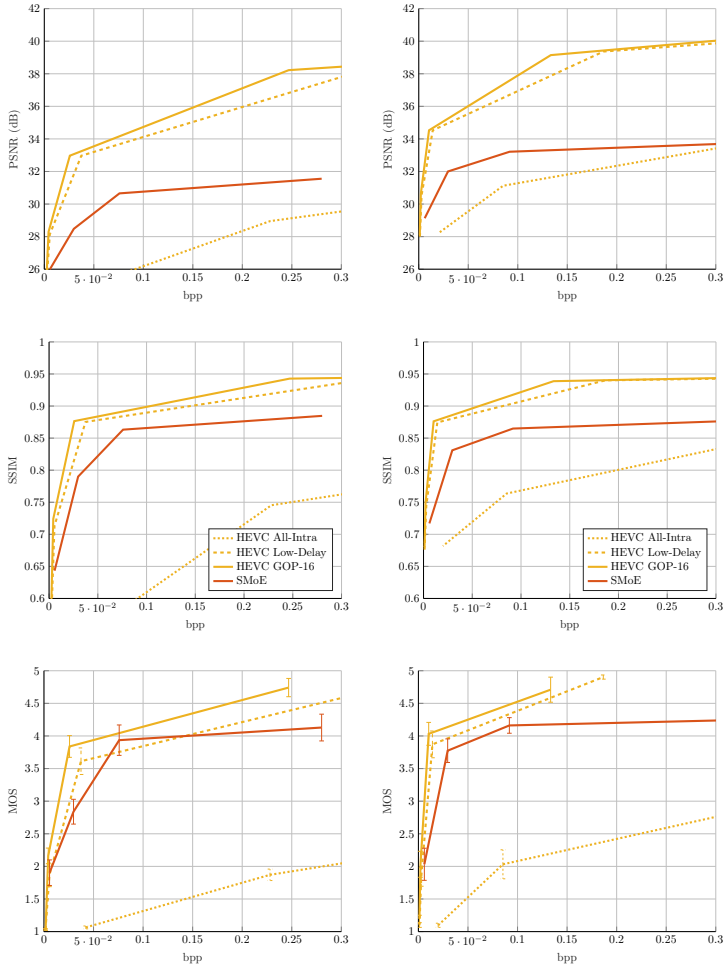
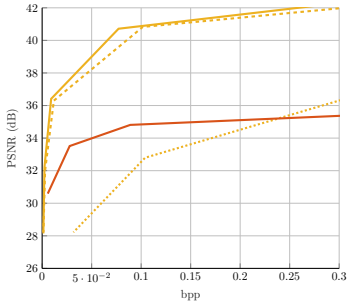
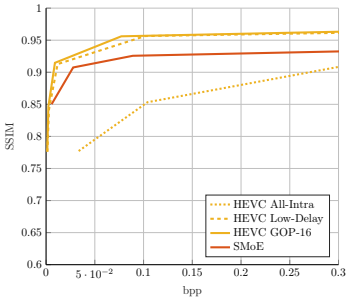


Figure 6.11: Coding results comparing the three HEVC (All-Intra, Low-Delay, and GOP-16) and SMoE in terms of PSNR, SSIM, and MOS scores for the light fields I03 Flowers *and* I04 Stone Pillars Outside.

(e) I10 Friends 1





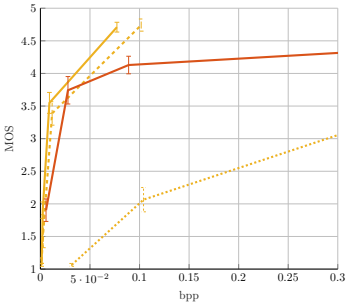
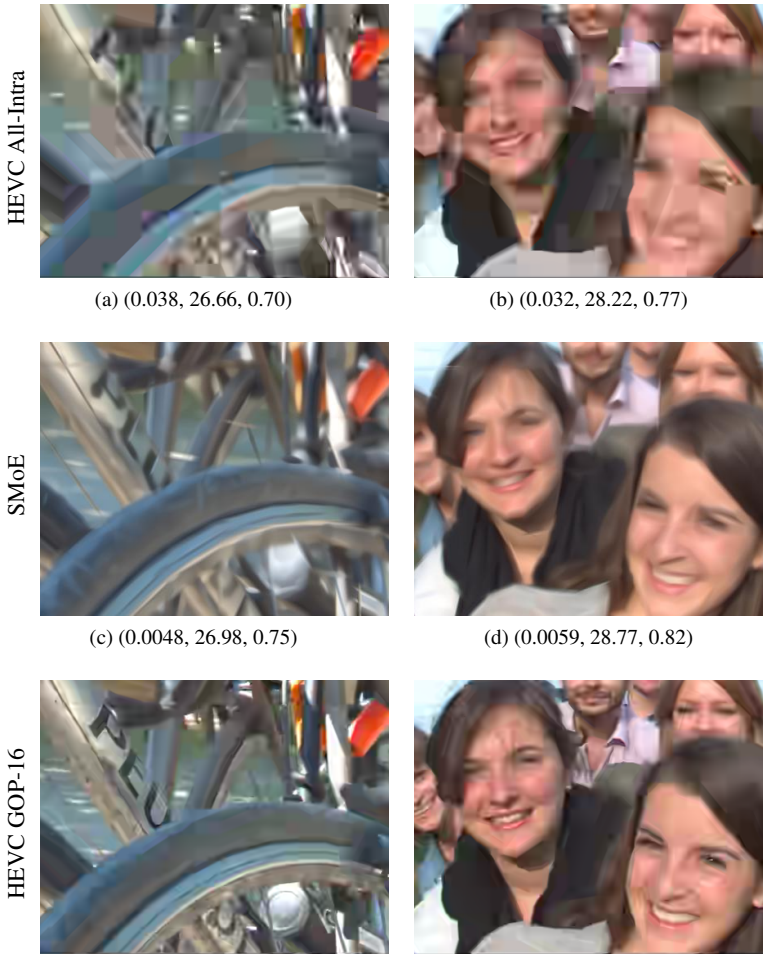


Figure 6.12: Coding results comparing the three HEVC (All-Intra, Low-Delay, and GOP-16) and SMoE in terms of PSNR, SSIM, and MOS scores for the final light field "I10 Friends 1". Notice in all above figures, the difference between the objective metrics (PSNR and SSIM) and subjective scores. It is clear that the distortions produced by SMoE (e.g. geometrical distortions, smoothing) are punished heavily by PSNR, and in lesser amount by SSIM. The SMoE distortions seem to have a visually pleasing effect as a MOS score of 4 indicates "Perceptual difference, but not annoying". Nonetheless, the loss of fine texture make it hard to achieve a MOS score closer to 5 "No perceptual difference". In general, we can say that up to and including a MOS score of 4, SMoE is competitive with motion-compensated pseudo-video coding of light fields using HEVC. Furthermore, note that PSNR and SSIM only capture the exact view reconstructions, whereas the MOS scores also captures smoothness between views and refocusing.



(a) (0.038, 26.66, 0.70) (b) (0.032, 28.22, 0.77)

(c) (0.0048, 26.98, 0.75) (d) (0.0059, 28.77, 0.82)

(e) (0.005, 30.36, 0.80) (f) (0.002, 32.21, 0.84)

Figure 6.13: Visual comparison of HEVC All-Intra (top row), SMoE (middle row), and HEVC GOP-16 (bottom row) for Bikes *(left) and* Friends *(right). For each illustration, the objective metrics are shown in the caption in the format (bits-per-pixel (bpp), PSNR (dB), SSIM). Bitrates are calculated as the LF filesize in bits divided by the number of pixels in the lenslet image [108].*

Figure 6.14: The setup used for the subjective experiments showing the 1080p Barco LC-47 monitor at eye-height. The light was turned off during the test. Both the ground truth and compressed sequences were shown side-by-side at native resolution.

coded, i.e. $2 \leq a_1 \leq 14$ and $2 \leq a_2 \leq 14$, as they are not used in calculating the objective metrics. Three configurations are compared: HEVC All-Intra (GOP=1), low-delay (GOP=4), and with GOP=16, with *Group-of-Pictures* (GOP) being the number of frames per single I-frame, ranging from granular random access (like SMoE) to low random access.

Fig. 6.10, Fig. 6.11, and Fig. 6.12 show the rate-distortion (RD) curves for five LFs: *I01*, *I02*, *I03*, *I04*, and *I10*, optimized to SSIM. Table 6.1 shows the parameters and the metrics for each RD-point. Three HEVC configurations with batch- and minibatch-based SMoE were compared. It is clear that for all images SMoE performs better than All-Intra HEVC with granular random access. However, SMoE is being consistently outperformed by motion-compensated HEVC. Batch and minibatch perform equally well, up until the point the batch-method does not allow higher kernel numbers. A visual comparison is provided in Fig. 6.13. For HEVC, this visual comparison confirms that higher amounts of views in between I-frames, yield better RD-performance. However, this increases the number of frames that need to be decoded in the worst case scenario and thus decreases the random access. On the other hand, having each frame as an I-frame results in bad RD-performance. Visually, we can see that SMoE provides an overly smooth representation at these low bit rates.

6.5.4 Subjective light field experiments

Subjective tests were performed in order to assess a more general quality of experience, which aims to capture view reconstruction quality, view consistency, and refocus quality simultaneously. The recommended guidelines on passive subjective evaluation of light fields were strictly followed as in [37]. *mean opinion scores* (MOS) were measured using a *double stimulus impairment scale* (DSIS), i.e. showing both the ground truth and the compressed sequence side-by-side.

Four RD-points of the three HEVC configurations and for the minibatch SMoE method were selected in the lowest range, as this was assumed to cover the highest variance in MOS scores. Eleven refocused images were calculated using the *LF-FiltShiftSum* function of the Matlab light field toolbox [23], the same slope values were used as suggested in Viola, Rerabek, and Ebrahimi [37].

The participant was not able to interact with the content, but a video was constructed for each RD-point that traverses the light field going through 97 selected viewpoints in a snake-like manner at 10 frames-per-second (fps) [37]. Next, the eleven refocused images were shown in an animation of 4 fps, going from a focused foreground to a focused background and back. The combined sequence in total was thus 15 seconds long. The participant was asked to rate the compressed sequence on a scale: 1 (Very Annoying), 2 (Annoying), 3 (Slightly annoying), 4 (Perceptible but not annoying), and 5 (Imperceptible).

The experiment was performed in two sessions. Each session showed 40 stimuli side-by-side (± 15 min per session) in a controlled environment as shown in Fig. 6.14. The monitor was a high-quality and color-calibrated Barco LC-47 at 1080p native resolution. The 30 subjects (24 male and 6 female of which 6 experts) were aged between 23 and 64 (mean 31).

Results are shown in the Figures 6.10, 6.11, and 6.12. Confidence intervals are plotted according to the ITU-R BT.500-13 recommendation [140]. It is very interesting to notice that subjectively SMoE scores much better compared to the objective metrics PSNR and SSIM, with PSNR differences up to 6dB. One explanation could be that the distortions introduced by SMoE (e.g. geometrical distortions and smoothing) are visually pleasing degradations. Furthermore, due to the continuous representation over all dimensions, SMoE is extremely view consistent. HEVC often introduced flickering when moving through views. The conclusion is that for MOS scores up to 4 (Perceptible but not annoying), SMoE coding is competitive with motion-compensated HEVC. However, a MOS score of 5 (Imperceptible) remains hard to achieve as our kernels fail to capture higher spatial frequencies.

6.5.5 Light field view consistency experiments

One desirable property in light field compression is that the the transitions from view to view are smooth. In this experiment, the angular consistency of the views is assessed based on two view traversals, both traversals start at view (3,3) and end at (11,11). One traverses the views horizontally, the other vertically. Both traversals traverse in a snake-like manner as can be seen in Fig. 6.15.

Let us introduce a novel metric in order to assess the amount of flickering when moving from one view to another. Both the original and the reconstructed LFs are traversed horizontally and vertically as described above. The original is subtracted from the reconstructed. From the residual traversal, DCT is performed along the

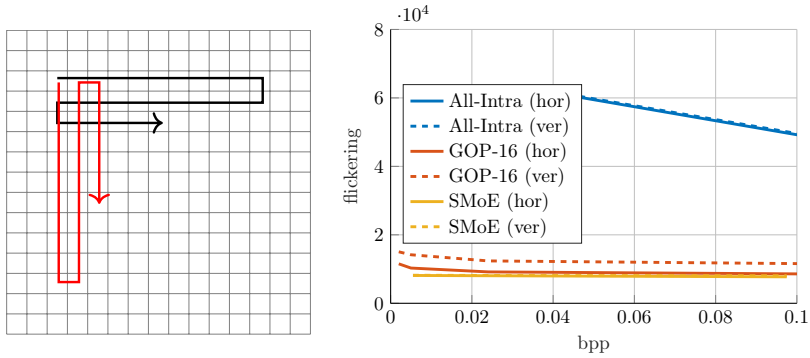


Figure 6.15: The two traversal options are illustrated on the left: horizontal (black) and vertical (red) traversal. On the right side, the traversal flickering is quantified for three LF coding methods: two using HEVC (All-Intra and GOP-16) and SMoE.

time dimension for each pixel in the residual. For each spatial coordinate, the sum of the last 10 absolute AC-coefficients are taken as indicator for the amount of view flickering at that pixel location. The mean of all pixel locations' flickering is then used as the amount of flickering for that traversal.

Fig. 6.15 shows that the amount for flickering is the lowest for SMoE, followed by HEVC GOP-16. The amount of flickering is high for HEVC All-Intra. Note that for HEVC All-Intra and SMoE the amount of flickering is equal for the horizontal and the vertical traversal, which is not the case for HEVC GOP-16. For HEVC GOP-16, the traversal contains the least flickering when it follows the same view traversal that was used during the coding (horizontal), as the views were based upon the last view seen. When traversing vertically, almost none of the subsequent views are in the same order used for coding the views. As such, the views jump from one part of the HEVC sequence to another, which introduces the flicker.

Subjectively, it is clear that HEVC All-Intra introduces heavy flickering. For low-bitrates, HEVC GOP-16 clearly introduces more flicker than SMoE. However, for higher bitrates both vertical and horizontal traversals do not show any flicker anymore. Interestingly enough, SMoE does not introduce flicker at all for both low and high bitrates. This is due to the steering of the kernels that explore long-range correlation between many frames of different views. Inter-view consistency is an in-built property of SMoE by design. The view traversal is pleasingly smooth in both traversals.

6.6 Light Field Video Coding

In Sec. 4.4, SMoE models were introduced for light field video. Light field videos are parametrized using a 5-D coordinate space (t, a_1, a_2, x_1, x_2) , i.e. $p = 5$. The

Figure 6.16: The three different light field video sequences used in this work. From left to right: cats - *512×352 (109 frames),* train1 - *512×352 (84 frames), and* train2 - *544×320 (97 frames).*

number of samples in a light field video becomes enormous even for very short, low-resolution videos. The dataset used here contains roughly 1 billion samples for ± 3 second clips. Nevertheless, in Sec. 5.3.6.3, it was shown how to model such large datasets in a feasible manner using the proposed progressive modeling approach. In Chapter 4, it was concluded that the SMoE (and sparse approaches in general) heavily becomes more interesting compared to dense representations when the dimensionality p of the coordinate space increases. The reason is that the average number of pixels covered by a single kernel grows exponentially in p. In this section, the goal is to investigate if this translates to coding gain using a similar coding approach as presented in the previous section.

In order to compare to the state of the art, the multiview extension of HEVC (MV-HEVC) is used as the anchor. Furthermore, different prediction structures were implemented with varying levels of random access capabilities. In this section, the same light field video dataset is used as in Sec. 4.4 and Sec. 5.3.6. For the three light field sequences *cats*, *train1* and *train2*, the coding performance was evaluated on a realistic MV-HEVC prediction scheme and compared them to the SMoE light field video coding approach [59], [115]. The dataset consists of two light field video sequences of resolution 512×352 and one at 544×320 at 30 fps for approximately 100 frames and 8×8 views, as illustrated in Fig. 6.16. Note that the sequences originate from temporally upscaling 4-D light fields originating from a lenslet-type camera, the frames thus include some artifacts from this process [115].

The coding approach is close to identical to the coding approach proposed for static 4-D light fields as illustrated in Fig. 6.7. There are two main differences in this section compared to the presented coding of static 4-D light fields. First, the models were built using the progressive modeling (Sec. 5.3.6) approach instead of the block-based modeling approach. Secondly, as p increases, the kernel parameters increase, i.e. $\boldsymbol{\mu}$ is now 8-D as well as the resulting DPCM prediction error vector e. Similarly, the coordinate covariance matrix is now 5-by-5, however this only increases the dimensionality of the codebook, while the number of parame-

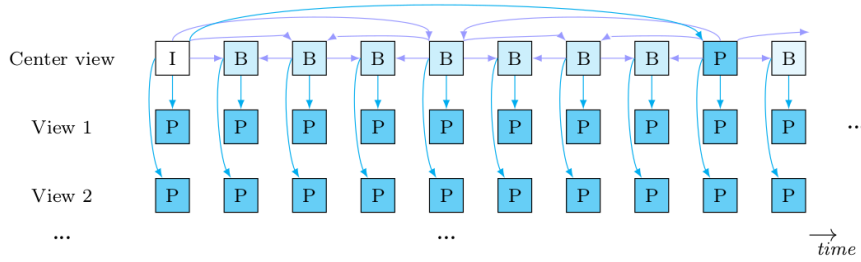


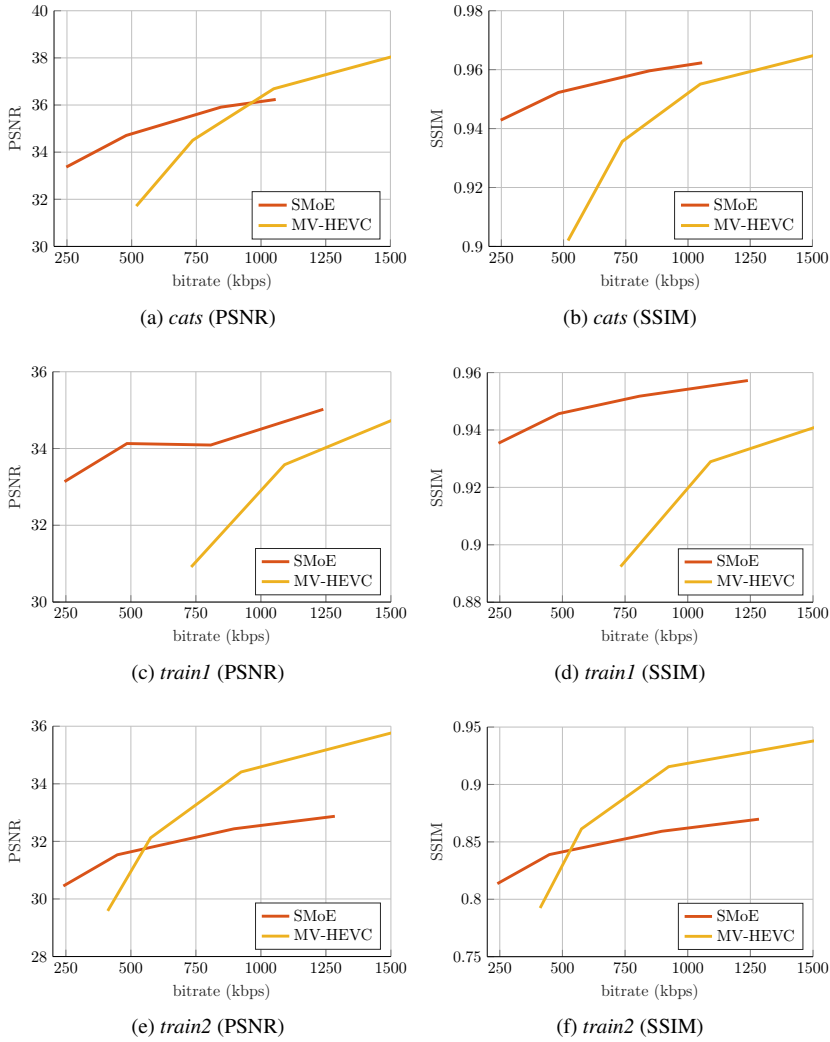
Figure 6.17: Illustration of the MV-HEVC configuration as proposed by Avramelos et al. [42]. Each center-view is predicted from other center views. However, within one frame, all views are predicted from the center-view.

ters per kernel does not increase: only the scale and the corresponding codebook entry identifier need to be stored.

6.6.1 Light field video coding experiment

In this experiment, the RD-performance of MV-HEVC is compared to our SMoE approach. Avramelos et al. investigated several MV-HEVC configurations for light field video encoding in terms of random access functionality and RD-performance. They propose a MV-HEVC configuration that includes only a single I-view. The configuration is illustrated in Fig. 6.17. The center-view of the first frame is the I-frame and each next center-view is predicted from other center-views [42]. Other views in a frame are then subsequently estimated from the center-view. This configuration thus results in a practically feasible solution in terms of random access behavior while only relying on a single I-frame for the entire light field video. In these experiments, the SMoE coding scheme is compared to the MV-HEVC configuration of Avramelos et al [42].

The SMoE models used in this experiment are the models that resulted from the progressive modeling experiments in Sec. 5.3.6. Fig. 6.18 illustrates the RD-curves in terms of PSNR and SSIM. Note, however, that SSIM correlated better with the subjective results from Sec. 6.5.4. Metrics are calculated on the three color planes and averaged as follows: $(6*\text{PSNR}_\text{Y}+\text{PSNR}_\text{Cb}+\text{PSNR}_\text{Cr})/8$, analogous for SSIM. It is clear that SMoE impressively outperforms MV-HEVC for the *cats* and *train1* sequences with bitrate savings up to a factor of 4x for the same quality. For the sequence *train2*, SMoE only outperforms MV-HEVC up to around 0.85 SSIM. The reason for this is that *train2* contains a lot of fine-grained texture that is hard to capture using our current model. Table 6.2 shows the parameters used for the SMoE encoding of the model parameters and the corresponding results in PSNR and SSIM. Additionally, Fig. 6.19 provides a subjective comparison. At low bitrates, SMoE results in spatio-temporal smoothing. In contrast, MV-



(a) *cats* (PSNR) (b) *cats* (SSIM)

(c) *train1* (PSNR) (d) *train1* (SSIM)

(e) *train2* (PSNR) (f) *train2* (SSIM)

Figure 6.18: Rate-distortion performance of MV-HEVC versus SMoE for three different light field video sequences. It is clear that SMoE provides high bitrate savings up to a factor of 4x for the sequences cats *and* train1. *For the sequence* train2*, MV-HEVC outperforms SMoE for quality levels over 32dB PSNR. The SMoE model is not able to adequately represent the many fine-grained textures, e.g. in the carpet.*

(a) 479.2 kbps: 34.70 dB PSNR, 0.952 SSIM

(b) 518.0 kbps: 31.71 dB PSNR, 0.902 SSIM

(c) 485.0 kbps: 34.12 dB PSNR, 0.946 SSIM

(d) 732.0 kbps: 30.91 dB PSNR, 0.892 SSIM

(e) 896.5 kbps, 32.44 dB PSNR, 0.859 SSIM

(f) 924.0 kbps, 34.41 dB PSNR, 0.915 SSIM

Figure 6.19: Subjective comparison of SMoE (left) vs MV-HEVC (right) of cats *(top),* train1 *(middle), and* train2 *(bottom) at frame 60. Metrics are indicated in PSNR and SSIM. SMoE is shown at respectively 7.5%, 33.7%, 3.0% less bitrate compared to MV-HEVC, while achieving superior objective quality around +3dB PSNR and +0.05 SSIM for the sequences* cats *and* train1. *However, MV-HEVC outperforms SMoE with 2dB and 0.06 SSIM for the* train2 *sequence.*

	kbps	K	L	b	σ_T	σ_A	σ_Y	σ_{UV}	σ_s	σ_{XY}	σ_{TY}	σ_{AY}	PSNR (dB)	SSIM
cats	247.487	7915	4096	12	0.12	0.12	0.50	0.25	0.25	0.25	0.25	0.25	33.37	0.943
	479.163	11454	4096	12	0.50	0.12	0.25	0.12	0.12	0.50	1.00	0.25	34.71	0.952
	845.481	23655	4096	12	0.50	0.12	0.50	0.12	0.25	1.00	0.25	0.50	35.91	0.960
	1056.801	33415	4096	12	0.12	0.12	0.25	0.12	0.25	0.50	1.00	0.25	36.24	0.962
train1	244.907	6475	1024	10	0.50	0.12	0.50	0.12	0.50	1.00	0.50	0.50	33.14	0.935
	484.981	9379	4096	10	0.50	0.12	0.25	0.25	0.50	0.50	0.50	1.00	34.13	0.946
	807.091	19230	4096	10	0.50	0.50	0.25	0.25	0.50	1.00	0.25	1.00	34.09	0.952
	1240.105	26954	4096	12	0.50	0.12	0.50	0.25	0.50	0.50	0.25	0.50	35.02	0.957
train2	240.386	7664	4096	10	0.50	0.50	0.25	0.25	0.25	1.00	0.25	0.25	30.45	0.814
	447.923	16115	4096	10	0.50	0.12	0.25	0.25	0.12	1.00	0.50	1.00	31.53	0.839
	896.473	23361	4096	12	0.12	0.12	0.25	0.12	0.50	0.50	0.50	0.25	32.44	0.859
	1284.378	33832	4096	12	0.50	0.12	0.50	0.12	0.50	0.50	1.00	0.25	32.87	0.870

Table 6.2: Coding parameters, objective (PSNR, SSIM) quality results for 5-D LF video SMoE models. The models contain K kernels. The covariance matrix R_{XX} is normalized by dividing the scale $s = |R_{XX}|^{1/p}$. The normalized R_{XX} is then coded using a dictionary with L entries. The identifier of the closest kernel in that dictionary is then coded using $\log_2(L)$ bits. The dictionary is trained on the set of normalized R_{XX} of this specific model and is transmitted along with the other parameters. All other parameters are normalized to have zero mean and a certain standard deviation σ depending on the parameter type. We then quantize using a fixed quantization step by dividing the maximum range into 2^b steps. All σ are ≤ 1. The rationale is that parameters with $\sigma < 1$ are quantized with more distortion, in order to save bits on these less important parameters. σ is 1, except for σ_T, σ_A, σ_Y, σ_{UV}, corresponding to the components of the prediction error $\mathbf{e}$ corresponding to time ($\mathbf{e}_T$), camera coordinates ($\mathbf{e}_A$), luma center ($\mathbf{e}_{Y_Y}$), and chroma centers ($\mathbf{e}_{Y_{Cb}}, \mathbf{e}_{Y_{Cr}}$). Next, σ_s, σ_{XY}, σ_{TY}, σ_{AY} are the σ-values for respectively the scale s per kernel, the covariance between the luma channel and spatial, time, and camera coordinates. All parameters are then arithmetic coded, assuming a Laplace distribution as initialization.

HEVC typically exhibits more blocking artifacts, e.g. visible in the horizontal lines behind the train and around the train chimney. Furthermore, SMoE exhibits in general better temporal consistency, especially in static segments. Such kernels have a long spread along the temporal dimension t and the camera coordinate plane (a_1, a_2). As such, these kernels yield consistent views. In such areas, extra kernels can thus be used to increase spatial detail instead of temporal detail. To conclude, the application of the SMoE representation becomes increasingly interesting as the dimensionality p of the coordinate space increases.

6.7 Conclusion

In this chapter, the application of coding using the SMoE model was investigated for 2-D images, 4-D LF images and 5-D LF video. The employed coding schemes followed the same general structure of (1) modeling the image modality (blockwise using kernel budgets or progressively), (2) quantizing the parameters per parameter type, and (3) entropy coding the quantized values. The coordinate covariance matrix R_{XX} was coded differently for the light field modalities compared

to 2-D images. For 2-D, the R_{XX} is robustly defined by the rotation of the main principle component and the two Eigen-values. However, such a scheme based on rotations is hard to generalize to higher dimensional modalities. Therefore, an effective vector quantization-like method is used based on the Kullback-Leibler divergence, which is scalable in dimensionality. Furthermore, redundancy between neighboring kernels was exploited by sorting the kernels along an approximation of the shortest path. Along this path, DPCM is performed to remove the redundancy in consecutive kernels in that path.

In the previous chapter, it was shown that the efficiency of SMoE increases when the dimensionality of the image modality increases. The reason is twofold. First, in dense representations the number of necessary pixels grows exponentially with the dimensionality. In contrast, sparse representations follow a more linear relationship depending on the image content. As a result, the average pixel-coverage by kernels increases exponentially as the dimensionality increases. Secondly, the number of parameters per kernel increases only quadratically when the dimensionality increases. Therefore, the assumption was that SMoE would be more competitive as a coding scheme as p increases. This assumption is proven to be correct in this chapter as indicated as follows.

For 2-D images, the proposed method outperformed JPEG for low bitrates, however, SMoE was consistently outperformed by JPEG-2000. From the results, it was clear that SMoE performs better on images that have varying spatial frequencies, compared to images that are mainly high-frequenced. For the latter, virtually every pixel is important and thus a purely spatial sparsification/clustering approach is less competitive compared to dense representations combined with DCT/wavelet transforms. For static 4-D light fields, the SMoE-based codec was outperformed by HEVC when using motion-compensation (low random access, complex decoding structure) in terms of PSNR and SSIM, the SMoE-based codec did strongly outperform HEVC All-Intra (which allows similar granular random access as SMoE). Subjective tests were performed in order to assess view quality, view consistency, and refocusing after coding. These results remarkably show that SMoE is competitive with the best HEVC configuration up to the range of a MOS score above 4 (Perceptible but not annoying), arguably the most interesting range for coding schemes from a practical point of view. For 5-D light field video, it was shown that SMoE can heavily outperform MV-HEVC up to bitrate savings up to a factor of 4x based on two LF videos. However, the SMoE method did not outperform MV-HEVC for higher qualities for one LF video sequence. Again, the reason is that the majority of the video consisted of high-frequenced texture.

To conclude, using SMoE as a coding approach becomes more competitive as the dimensionality of the coordinate space p increases as expected. Additionally, all the beneficial properties of SMoE are still present when using this coding scheme. The decoded model thus still obtains a continuous representation that can

be sampled in parallel at pixel level. Furthermore, the model provides low-level descriptors for the post-processing tasks. However, the SMoE coding method is outperformed by other coding schemes when the images mainly contains fine-grained textures. Even though, it is important to note that subjectively the SMoE method performs much better than compared to using objective metrics. Subjectively, the lack of fine texture seems to be less annoying. Nevertheless, it is surely beneficial to explore the use of more expressive and equally sparse models for representing any-dimensional image modalities.

7 Conclusion and Future Work

7.1 Conclusion

In this dissertation, a radically novel any-dimensional image representation method was proposed. The goal was to design a method that in the future could enable wide-range 6-DoF virtual reality based on camera-captured content. This work provided evidence of the feasibility of designing a sparse information-rich representation for such goals. The application in focus was the coding of immersive image modalities, however, the possible applications of the proposed model are not limited to coding. The presented model scales to any dimensionality while enabling desired functionality for VR consumption, e.g. random access, inherent view interpolation, and pixel-parallel reconstructions.

In Chapter 1, the concepts and challenges of camera-captured 6-DoF VR were discussed. Additionally, the technological solutions that are being pursued at the moment were evaluated, e.g. 3-D construction of a scene and using methods from traditional video-coding. Furthermore, the desired functionalities and requirements for a 6-DoF representation were identified. The problem of camera-captured VR was broken down to the most fundamental theoretical concepts in Chapter 2, e.g. the plenoptic function and light fields. Light fields were identified to be an extremely interesting concept. However, some practical challenges in working with light fields were identified. The current challenges are mainly related to the enormous amounts of data that is commonly present in light fields, as well as the limited capturing devices at the moment.

The SMoE representation was consecutively introduced in Chapter 3. First, the method was discussed from a theoretical viewpoint and then further illustrated on 2-D image examples. It was shown how an image modality can be described using a sparse statistical model using spatial kernels. Furthermore, it was illustrated how such a data-adaptive model exhibits the data structure and reveals image descriptors through the kernel parameters. In Chapter 4, the SMoE representation has the property of being scalable towards higher-dimensional image modalities. The model was thus applied and discussed for some immersive image modalities, e.g. omnidirectional images, 4-D LF images and 5-D LF video. The efficient use of SMoE on spherical dimensions was proven for omnidirectional images. In the case of light fields, it was illustrated how the kernels follow the spatial structure of the pixel data as the kernels elongate along the EPI structures and the optical flow in the time dimension. Additionally, it was shown how the model's kernels can capture up to 10,000s of original pixels in light field video.

A secondary, albeit crucial contribution of this work was presented in Chapter 5: the computationally efficient modeling of extremely large datasets using thousands of distributions. Despite the bad theoretical complexity of the employed modeling algorithms, it was shown how the complexity can be contained by exploiting the local structure and relying on the conditional computation principle in MoEs. Finally, in Chapter 6, a coding scheme was devised that quantizes and binarizes the model parameters in order to efficiently transmit and store SMoE models. It was shown how the RD-performance of the coding scheme improves as the dimensionality of the image modalities increases. For 2-D images, only limited gains were possible using the current modeling strategies. While for light field images, the performance was subjectively close to the state of the art. Finally, for light field video, our coding scheme can even heavily outperform the state of the art on two of the three tested videos. Moreover, our model yields additional functionality at decoding side.

Nevertheless, the SMoE model is faced with challenges when modeling high frequenced image data, e.g. fine-grained spatial texture or temporal flickering. The representation employs only linear regressors and thus assumes natural images to be able to be approximated as a smoothed piecewise linear function. However, the reality is that image modalities resemble more piecewise stationary functions and can exhibit high spatial frequencies in textured regions. With the current model and employed optimization strategies, an infeasible number of small kernels would be necessary to capture all detail.

7.2 Future work

Light fields allow for 6-DoF user experience only if the "open space" assumption is met. It is therefore not possible to walk in between objects. For a full 6-DoF

experience, the full plenoptic function, including its spherical dimensions should be modeled. In this work, it was shown that our model can represent a mixture of Euclidean and spherical coordinate dimensions. Thus, in theory, the whole plenoptic function could be modeled. However, some practical challenges remain in terms of acquisition and modeling even more extreme data sets. Modeling the plenoptic function is thus considered future work, although this dissertation did ensure theoretical compatibility.

Future work on increasing the RD-performance of the SMoE framework for all image modalities lays in the model design, as well as in the modeling process. First, the model could be designed to e.g. incorporate residual texture. Secondly, the optimization process, i.e. finding the optimal kernel parameters has a big impact on the reconstruction quality. In this work, the model does not maximize the PSNR of the reconstruction, but maximizes the likelihood of the model. As such, PSNR optimization could be a way to increase the RD-performance as early evidence shows [141], [142]. Very recently, Jongebloed et al. have shown impressive coding results on 2-D images, even outperforming JPEG with 42% when performing in-loop quantization during modeling [143]. These examples show how important the optimization process in finding the optimal kernel parameters is. Much gain remains present, possibly even without changing the model.

The secondary functionalities that were not fully investigated in this dissertation also remain future work. These functionalities include, denoising, super-resolution, segmentation, ... of all possible image modalities. Especially the link between the depth of a scene and the kernel parameters need to be further exploited as it could potentially leverage depth-dependent processing tasks.

In general, it is my opinion that the evaluation of immersive modalities needs much future investigation. The evaluation of methods need to be reviewed in terms of (1) reconstruction quality, (2) defining rate, (3) evaluating secondary functionality, and (4) complexity. Furthermore, the maturity of the methods are ideally taken into account.

Firstly, let us consider visual quality metrics. For example, the usage of metrics such as PSNR and SSIM are questionable for the following reasons. First, in the past, progress in video coding was made on hybrid MC-transform methods. It is long known that there are issues with PSNR and SSIM as quality metrics. However, it made some sense to compare using the PSNR and SSIM metrics as the artifacts among technologies were similar of nature. The question remains if it is sensible when using a fundamentally different methods with artifacts that are completely different in nature, e.g. geometrical distortions. Evidence for these doubts are shown in Chapter 4 for static light fields in which the subjective results are much more in favor of SMoE compared to PSNR and SSIM. Second, only the reconstruction quality of the captured views were measured using PSNR and SSIM. Whereas, the main application for light field is light field rendering, in

which non-captured views are reconstructed. Consistency between the views is crucial for light field rendering. The performance of light field rendering is thus not measured. Note that, the application of refocusing was incorporated in the subjective tests.

Secondly, the term "rate" is ill-defined in immersive applications as illustrated by the following two examples. First, consider light field images taken using a lenslet-type camera. Is rate defined by the number of pixels on the original lenslet image, or by the sum of derived pixels in the subaperture images? Different lenslet-processing methods can result in different subaperture resolutions. Secondly, consider streaming light field video, does it make sense to have the rate always include all possible views at decoder side? This is not how light fields will be used per se. When performing light field rendering or refocusing, pixels are combined from multiple views simultaneously. In these cases, the interest lays the relevant pixels from each view.

Thirdly, it is extremely difficult to compare secondary functionality, e.g. random access, between methods that are fundamentally different in nature. For example, once all necessary views are buffered in 3D-HEVC, then a range of views can be rendered independently using view synthesis. However, is the view synthesis process really a part of the coding scheme or a post-processing method? In SMoE, once all relevant kernel parameters are decoded, then all views can be rendered based on these parameters. However, the parameter decoding is still linear in nature in our proposed coding scheme. The same holds for HEVC-like methods as they also depend on a arithmetic coder which is inherently linear.

Fourthly, care is needed when comparing the complexity of methods that are at different levels of maturity. Rendering methods from 3D-graphics are by itself extremely complex. However, complex rendering techniques have become feasible due to the high level of maturity and specific hardware design. Similarly, coding methods such as HEVC contain processes that are actually extremely bad in terms of theoretical complexity, e.g. finding motion vectors. However, the implementations have become remarkably efficient due to the level of maturity.

To conclude, it is in the community's best interest to further develop standardized evaluation frameworks that take into account the novel paradigms in immersive applications in a more holistic fashion. It is clear that there is still a long way to go before we can enjoy the first full wide-range CC-VR applications. Nevertheless, this work provides a scalable framework that is future-proof to facilitate the road towards full CC-VR while showing alternative and competitive methods for representing and coding more common image modalities as well. In general, this work is not meant in any way to be the final solution, but to open a novel way of thinking in the development of image representations and to abandon some potentially outdated paradigms.

Bibliography

[1] A. T. Hinds, D. Doyen, and P. Carballeira, "Toward the realization of six degrees-of-freedom with compressed light fields", in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2017, pp. 1171–1176, ISBN: 978-1-5090-6067-2. DOI: 10.1109/ICME.2017.8019543.

[2] id Software. (2017). Wolfenstein 3D, [Online]. Available: https://en.wikipedia.org/wiki/Wolfenstein_3D (visited on 07/04/2017).

[3] S. Orts-Escolano, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S. B. Kang, C. Rhemann, P. Kohli, Y. Lutchyn, C. Keskin, S. Izadi, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, and S. Khamis, "Holoportation: Virtual 3D Teleportation in Real-time", in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*, New York, New York, USA: ACM Press, 2016, pp. 741–754, ISBN: 9781450341899. DOI: 10.1145/2984511.2984517.

[4] M. Mori, K. F. MacDorman, and N. Kageki, "The uncanny valley", *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 98–100, 2012, ISSN: 10709932. DOI: 10.1109/MRA.2012.2192811.

[5] M. Domanski, O. Stankiewicz, K. Wegner, and T. Grajek, "Immersive visual media — MPEG-I: 360 video, virtual navigation and beyond", in *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*, IEEE, 2017, pp. 1–9, ISBN: 978-1-5090-6344-4. DOI: 10.1109/IWSSIP.2017.7965623.

[6] L. Yu, G. Lafruit, J. Jung, B. Kroon, and J. Boyce, "AhG report on MPEG-I Visual Technologies", Moving Pictures Experts Group (MPEG), Geneva, Switzerland, Tech. Rep., 2019.

[7] H. Shum and S. B. Kang, "Review of image-based rendering techniques", K. N. Ngan, T. Sikora, and M.-T. Sun, Eds., 2000, p. 2. DOI: 10.1117/12.386541.

[8] G. Tech, Y. Chen, K. Muller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35–49, 2016, ISSN: 1051-8215. DOI: 10.1109/TCSVT.2015.2477935.

[9] T. Milliron, C. Szczupak, and O. Green, "Hallelujah", in *ACM SIGGRAPH 2017 VR Village on - SIGGRAPH '17*, New York, New York, USA: ACM Press, 2017, pp. 1–2, ISBN: 9781450350136. DOI: 10.1145/3089269.3089283.

[10] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What Will 5G Be?", *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014, ISSN: 0733-8716. DOI: 10.1109/JSAC.2014.2328098.

[11] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard", *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001, ISSN: 10535888. DOI: 10.1109/79.952804.

[12] G. Hudson, A. Leger, B. Niss, and I. Sebestyen, "JPEG at 25: Still Going Strong", *IEEE MultiMedia*, vol. 24, no. 2, pp. 96–103, 2017, ISSN: 1070-986X. DOI: 10.1109/MMUL.2017.38.

[13] I. Ihrke, J. Restrepo, and L. Mignard-Debise, "Principles of Light Field Imaging: Briefly revisiting 25 years of research", *IEEE Signal Processing Magazine*, vol. 33, no. 5, pp. 59–69, 2016, ISSN: 1053-5888. DOI: 10.1109/MSP.2016.2582220.

[14] K. Aizawa and T. S. Huang, "Model-based image coding advanced video coding techniques for very low bit-rate applications", *Proceedings of the IEEE*, vol. 83, no. 2, pp. 259–271, 1995.

[15] T. Sikora, "The MPEG-7 Visual Standard for Content Description – An Overview", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 696–702, 2001, ISSN: 10518215. DOI: 10.1109/76.927422.

[16] ——, "Trends and Perspectives in Image and Video Coding", *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6–17, 2005, ISSN: 0018-9219. DOI: 10.1109/JPROC.2004.839601.

[17] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012, ISSN: 1051-8215. DOI: 10.1109/TCSVT.2012.2221191.

[18] V. Avramelos, R. Verhack, I. Saenen, G. Van Wallendael, B. Goossens, and P. Lambert, "Highly parallel steered mixture-of-experts rendering at pixel-level for image and light field data", *Journal of Real-Time Image Processing*, 2018, ISSN: 1861-8200. DOI: 10.1007/s11554-018-0843-3.

[19] M. Levoy and P. Hanrahan, "Light field rendering", in *Proc. Conf. on Computer Graphics and Interactive Techniques - SIGGRAPH '96*, New York, New York, USA: ACM Press, 1996, pp. 31–42, ISBN: 0897917464. DOI: 10.1145/237170.237199.

[20] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light Field Image Processing: An Overview", *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, 2017, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2017.2747126.

[21] E. Adelson and J. Bergen, *The plenoptic function and the elements of early vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991.

[22] T. Oo, H. Kawasaki, Y. Ohsawa, and K. Ikeuchi, "Separation of Reflection and Transparency Using Epipolar Plane Image Analysis", in, 2006, pp. 908–917. DOI: 10.1007/11612032_91.

[23] D. G. Dansereau. (2015). Light Field Toolbox for Matlab, [Online]. Available: https://nl.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4 (visited on 06/10/2016).

[24] A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields", in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, New York, New York, USA: ACM Press, 2000, pp. 297–306, ISBN: 1581132085. DOI: 10.1145/344779.344929.

[25] M. Tanimoto, "FTV: Free-viewpoint Television", *Signal Processing: Image Communication*, vol. 27, no. 6, pp. 555–570, 2012, ISSN: 09235965. DOI: 10.1016/j.image.2012.02.016.

[26] T. Georgiev, Z. Yu, A. Lumsdaine, and S. Goma, "Lytro camera technology: theory, algorithms, performance analysis", C. G. M. Snoek, L. S. Kennedy, R. Creutzburg, D. Akopian, D. Wüller, K. J. Matherson, T. G. Georgiev, and A. Lumsdaine, Eds., 2013, 86671J. DOI: 10.1117/12.2013581.

[27] I. Hamilton. (2018). SIGGRAPH 2018: Learn About Google's Efforts To Capture Light Fields, [Online]. Available: https://uploadvr.com/siggraph-2018-learn-about-googles-efforts-to-capture-light-fields/ (visited on 08/14/2019).

[28] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong, "Plenoptic sampling", in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, New York, New York, USA: ACM Press, 2000, pp. 307–318, ISBN: 1581132085. DOI: 10.1145/344779.344932.

[29] Z. Lin and H.-Y. Shum, "A Geometric Analysis of Light Field Rendering", *International Journal of Computer Vision*, vol. 58, no. 2, pp. 121–138, 2004, ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000015916.91741.27.

[30] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured Lumigraph Rendering", in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01, New York, NY, USA: ACM, 2001, pp. 425–432, ISBN: 1-58113-374-X. DOI: 10.1145/383259.383309.

[31] B. Ceulemans, S. P. Lu, G. Lafruit, and A. Munteanu, "Robust Multiview Synthesis For Wide-Baseline Camera Arrays", *IEEE Transactions on Multimedia*, 2018, ISSN: 15209210. DOI: 10.1109/TMM.2018.2802646.

[32] R. Farrugia and C. Guillemot, "Light Field Super-Resolution using a Low-Rank Prior and Deep Convolutional Neural Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2019.2893666.

[33] A. Chuchvara, A. Barsi, and A. Gotchev, "Fast and Accurate Depth Estimation from Sparse Light Fields", 2018. arXiv: 1812.06856.

[34] Y. Li, M. Sjostrom, R. Olsson, and U. Jennehag, "Efficient intra prediction scheme for light field image compression", in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 539–543, ISBN: 978-1-4799-2893-4. DOI: 10.1109/ICASSP.2014.6853654.

[35] L. F. R. Lucas, C. Conti, P. Nunes, L. D. Soares, N. M. M. Rodrigues, C. L. Pagliari, E. A. B. da Silva, and S. M. M. de Faria, "Locally linear embedding-based prediction for 3D holoscopic image coding using HEVC", in *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, IEEE, 2014, pp. 11–15.

[36] C. Perra and P. Assuncao, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement", in *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, 2016, pp. 1–4, ISBN: 978-1-5090-1552-8. DOI: 10.1109/ICMEW.2016.7574671.

[37] I. Viola, M. Rerabek, and T. Ebrahimi, "Comparison and Evaluation of Light Field Image Coding Approaches", *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1092–1106, 2017, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2017.2740167.

[38] L. Li, Z. Li, B. Li, D. Liu, and H. Li, "Pseudo Sequence Based 2-D Hierarchical Coding Structure for Light-Field Image Compression", in *2017 Data Compression Conference (DCC)*, IEEE, 2017, pp. 131–140, ISBN: 978-1-5090-6721-3. DOI: 10.1109/DCC.2017.10.

[39] C. Conti, L. Ducla Soares, and P. Nunes, "Light Field Coding with Field of View Scalability and Exemplar-Based Inter-Layer Prediction", *IEEE Transactions on Multimedia*, pp. 1–1, 2018, ISSN: 1520-9210. DOI: 10.1109/TMM.2018.2825882.

[40] W. Ahmad, R. Olsson, and M. Sjostrom, "Interpreting plenoptic images as multi-view sequences for improved compression", in *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, pp. 4557–4561, ISBN: 978-1-5090-2175-8. DOI: 10.1109/ICIP.2017.8297145.

[41] H. Amirpour, A. Pinheiro, M. Pereira, F. Lopes, and M. Ghanbari, "Light Field Image Compression with Random Access", in *2019 Data Compression Conference (DCC)*, IEEE, 2019, pp. 553–553, ISBN: 978-1-7281-0657-1. DOI: 10.1109/DCC.2019.00065.

[42] V Avramelos, J De Praeter, G Van Wallendael, and P Lambert, "Random access prediction structures for light field coding with MV-HEVC", *Submitted to Springer Multimedia Tools and Applications*, 2019.

[43] J. Chen, J. Hou, and L.-P. Chau, "Light Field Compression With Disparity-Guided Sparse Coding Based on Structural Key Views", *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 314–324, 2018, ISSN: 1057-7149. DOI: 10.1109/TIP.2017.2750413.

[44] J. Zhao, P. An, X. Huang, C. Yang, and L. Shen, "Light Field Image Compression via CNN-Based EPI Super-Resolution and Decoder-Side Quality Enhancement", *IEEE Access*, pp. 1–1, 2019, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2930644.

[45] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens, "JPEG Pleno: Toward an Efficient Representation of Visual Reality", *IEEE MultiMedia*, vol. 23, no. 4, pp. 14–20, 2016, ISSN: 1070-986X. DOI: 10.1109/MMUL.2016.64.

[46] M. B. de Carvalho, M. P. Pereira, G. Alves, E. A. B. da Silva, C. L. Pagliari, F. Pereira, and V. Testoni, "A 4D DCT-Based Lenslet Light Field Codec", in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, pp. 435–439, ISBN: 978-1-4799-7061-2. DOI: 10.1109/ICIP.2018.8451684.

[47] R. A. Farrugia and J. A. Briffa, "Lossless Light Field Compression Using 4D Wavelet Transforms", in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 121–125, ISBN: 978-1-5386-6249-6. DOI: 10.1109/ICIP.2019.8802937.

[48] I. Schiopu and A. Munteanu, "Macro-Pixel Prediction Based on Convolutional Neural Networks for Lossless Compression of Light Field Images", in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, pp. 445–449, ISBN: 978-1-4799-7061-2. DOI: 10.1109/ICIP.2018.8451731.

[49] A. Alperovich, O. Johannsen, M. Strecke, and B. Goldluecke, "Light Field Intrinsics With a Deep Encoder-Decoder Network", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9145–9154.

[50] M. Rizkallah, X. Su, T. Maugey, and C. Guillemot, "Graph-based Transforms for Predictive Light Field Compression based on Super-Pixels", in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 1718–1722, ISBN: 978-1-5386-4658-8. DOI: 10.1109/ICASSP.2018.8462288.

[51] M. Hog, N. Sabater, and C. Guillemot, "Superrays for Efficient Light Field Processing", *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1187–1199, 2017, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2017.2738619.

[52] X. Su, M. Rizkallah, T. Mauzev, and C. Guillemot, "Rate-Distortion Optimized Super-Ray Merging for Light Field Compression", in *2018 26th European Signal Processing Conference (EUSIPCO)*, IEEE, 2018, pp. 1850–1854, ISBN: 978-9-0827-9701-5. DOI: 10.23919/EUSIPCO.2018.8553485.

[53] E. Dib, M. Le Pendu, X. Jiang, and C. Guillemot, "Super-Ray Based Low Rank Approximation for Light Field Compression", in *2019 Data Compression Conference (DCC)*, IEEE, 2019, pp. 369–378, ISBN: 978-1-7281-0657-1. DOI: 10.1109/DCC.2019.00045.

[54] R. Verhack, T. Sikora, L. Lange, G. Van Wallendael, and P. Lambert, "A universal image coding approach using sparse steered Mixture-of-Experts regression", in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, pp. 2142–2146, ISBN: 978-1-4673-9961-6. DOI: 10.1109/ICIP.2016.7532737.

[55] R. Verhack, N. Madhu, G. Van Wallendael, P. Lambert, and T. Sikora, "Steered Mixture-of-Experts Approximation of Spherical Image Data", in *2018 26th European Signal Processing Conference (EUSIPCO)*, IEEE, 2018, pp. 256–260, ISBN: 978-9-0827-9701-5. DOI: 10.23919/EUSIPCO.2018.8553065.

[56] L. Lange, R. Verhack, and T. Sikora, "Video representation and coding using a sparse steered mixture-of-experts network", in *2016 Picture Coding Symposium (PCS)*, IEEE, 2016, pp. 1–5, ISBN: 978-1-5090-5966-9. DOI: 10.1109/PCS.2016.7906369.

[57] R. Verhack, T. Sikora, L. Lange, R. Jongebloed, G. Van Wallendael, and P. Lambert, "Steered mixture-of-experts for light field coding, depth estimation, and processing", in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2017, pp. 1183–1188, ISBN: 978-1-5090-6067-2. DOI: 10.1109/ICME.2017.8019442.

[58] R. Verhack, G. Van Wallendael, M. Courteaux, P. Lambert, and T. Sikora, "Progressive Modeling of Steered Mixture-of-Experts for Light Field Video Approximation", in *2018 Picture Coding Symposium (PCS)*, IEEE, 2018, pp. 268–272, ISBN: 978-1-5386-4160-6. DOI: 10.1109/PCS.2018.8456242.

[59] V. Avramelos, I. Saenen, R. Verhack, G. Van Wallendael, P. Lambert, and T. Sikora, "Steered mixture-of-experts for light field video coding", in *Applications of Digital Image Processing XLI*, A. G. Tescher, Ed., SPIE, 2018, p. 11, ISBN: 9781510620759. DOI: 10.1117/12.2320563.

[60] R. Verhack, T. Sikora, G. Van Wallendael, and P. Lambert, "Steered Mixture-of-Experts for Light Field Images and Video: Representation and Coding", *IEEE Transactions on Multimedia*, pp. 1–1, 2019, ISSN: 1520-9210. DOI: 10.1109/TMM.2019.2932614.

[61] I. P. Saenen, R. Verhack, V. Avramelos, G. Van Wallendael, and P. Lambert, "Hard Real-Time, Pixel-Parallel Rendering of Light Field Videos Using Steered Mixture-of-Experts", in *2018 Picture Coding Symposium (PCS)*, IEEE, 2018, pp. 337–341, ISBN: 978-1-5386-4160-6. DOI: 10.1109/PCS.2018.8456306.

[62] R. Verhack, S. Van De Keer, G. Van Wallendael, T. Sikora, and P. Lambert, "Color prediction in image coding using Steered Mixture-of-Experts", in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, pp. 1288–1292, ISBN: 978-1-5090-4117-6. DOI: 10.1109/ICASSP.2017.7952364.

[63] S. Takagi, K. Sugiyama, S. Ii, and Y. Matsumoto, "A Review of Full Eulerian Methods for Fluid Structure Interaction Problems", *Journal of Applied Mechanics*, vol. 79, no. 1, p. 010 911, 2012, ISSN: 00218936. DOI: 10.1115/1.4005184.

[64] E. J. Candès and D. L. Donoho, "Curvelets: a surprisingly effective nonadaptive representation of objects with edges", in *In Curve and Surface Fitting: Saint-Malo*, University Press, 2000, pp. 0–82 651 357.

[65] G. Wallace, "The JPEG still picture compression standard", *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992, ISSN: 00983063. DOI: 10.1109/30.125072.

[66] P. Salembier, F. Marques, M. Pardas, J. Morros, I. Corset, S. Jeannin, L. Bouchard, F. Meyer, and B. Marcotegui, "Segmentation-based video coding system allowing the manipulation of objects", *IEEE Trans. Circuits Syst. Video Technol*, vol. 7, no. 1, pp. 60–74, 1997, ISSN: 10518215. DOI: 10.1109/76.554418.

[67] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video", *IEEE Trans. Circuits Syst. Video Technol*, vol. 5, no. 1, pp. 59–62, 1995, ISSN: 10518215. DOI: 10.1109/76.350781.

[68] E. Strekalovskiy and D. Cremers, "Real-Time Minimization of the Piecewise Smooth Mumford-Shah Functional", in, Springer, Cham, 2014, pp. 127–141. DOI: 10.1007/978-3-319-10605-2_9.

[69] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems", *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989, ISSN: 00103640. DOI: 10.1002/cpa.3160420503.

[70] P. Prandoni and M. Vetterli, "Approximation and compression of piecewise smooth functions", *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 357, no. 1760, B. W. Silverman and J. C. Vassilicos, Eds., pp. 2573–2591, 1999, ISSN: 1471-2962. DOI: 10.1098/rsta.1999.0449.

[71] H. Takeda, "Kernel Regression for Image Processing and Reconstruction", PhD thesis, University of California, Santa Cruz, 2006.

[72] R. Verhack, A. Krutz, P. Lambert, R. Van de Walle, and T. Sikora, "Lossy image coding in the pixel domain using a sparse steering kernel synthesis approach", in *2014 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2014, pp. 4807–4811, ISBN: 978-1-4799-5751-4. DOI: 10.1109/ICIP.2014.7025974.

[73] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks", DTIC Document, Tech. Rep., 1988.

[74] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression", *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004, ISSN: 0960-3174. DOI: 10.1023/B:STCO.0000035301.49549.88.

[75] J. Moody and C. Darken, *Learning with localized receptive fields*. Yale Univ., Department of Computer Science, 1988.

[76] R. D. De Veaux, "Mixtures of linear regressions", *Computational Statistics & Data Analysis*, vol. 8, no. 3, pp. 227–245, 1989, ISSN: 01679473. DOI: 10.1016/0167-9473(89)90043-1.

[77] S. Faria and G. Soromenho, "Fitting mixtures of linear regressions", *Journal of Statistical Computation and Simulation*, vol. 80, no. 2, pp. 201–225, 2010, ISSN: 0094-9655. DOI: 10.1080/00949650802590261.

[78] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive Mixtures of Local Experts", *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991, ISSN: 0899-7667. DOI: 10.1162/neco.1991.3.1.79.

[79] L. Xu, M. I. Jordan, and G. E. Hinton, "An alternative model for mixtures of experts", *Advances in neural information processing systems*, pp. 633–640, 1995.

[80] L. Ungar and R. De Veaux, "EMRBF: a statistical basis for using radial basis functions for process control", in *Proceedings of 1995 American Control Conference - ACC'95*, vol. 3, American Autom Control Council, 1995, pp. 1872–1876, ISBN: 0-7803-2445-5. DOI: 10.1109/ACC.1995.531211.

[81] I. Cha and S. Kassam, "RBFN Restoration of Nonlinearly Degraded Images", *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 964–975, 1996, ISSN: 10577149. DOI: 10.1109/83.503912.

[82] G. Bugmann, "Normalized Gaussian Radial Basis Function networks", *Neurocomputing*, vol. 20, no. 1-3, pp. 97–110, 1998, ISSN: 09252312. DOI: 10.1016/S0925-2312(98)00027-7.

[83] D. Kai Tik Chow and Tong Lee, "Image approximation and smoothing by support vector regression", in *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*, IEEE, 2001, pp. 2427–2432, ISBN: 0-7803-7044-9. DOI: 10.1109/IJCNN.2001.938747.

[84] H. Sung, "Gaussian Mixture Regression and Classification", PhD thesis, Rice University, 2004.

[85] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[86] H. Greenspan, J. Goldberger, and A. Mayer, "Probabilistic space-time video modeling via piecewise gmm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 384–396, 2004, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2004.1262334.

[87] G. Yazbek, C. Mokbel, and G. Chollet, "Video Segmentation and Compression using Hierarchies of Gaussian Mixture Models", in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, IEEE, 2007, pp. I–1009–I–1012, ISBN: 1-4244-0727-3. DOI: 10.1109/ICASSP.2007.366081.

[88] M. Hog, N. Sabater, and C. Guillemot, "Dynamic Super-Rays for Efficient Light Field Video Processing", in *BMVC 2018 - 29th British Machine Vision Conference*, Newcastle upon Tyne, United Kingdom, 2018, pp. 1–12.

[89] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty Years of Mixture of Experts", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, 2012, ISSN: 2162-237X. DOI: 10.1109/TNNLS.2012.2200299.

[90] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1994, ISBN: 0023527617.

[91] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer", {*arXiv preprint arXiv:1701.06538*}, 2017. arXiv: 1701.06538.

[92] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach", in *Advances in neural information processing systems*, J. D. Cowan, G Tesauro, and J Alspector, Eds., Morgan-Kaufmann, 1994, pp. 120–127.

[93] M. Jordan and L. Xu, "Convergence Results for the EM Approach to Mixtures of Experts Architectures", *Neural Networks*, vol. 8, no. 9, pp. 1409–1431, 1995, ISSN: 08936080. DOI: 10.1016/0893-6080(95)00014-3.

[94] S.-K. Ng and G. McLachlan, "Using the EM Algorithm to Train Neural Networks: Misconceptions and a New Algorithm for Multiclass Classification", *IEEE Transactions on Neural Networks*, vol. 15, no. 3, pp. 738–749, 2004, ISSN: 1045-9227. DOI: 10.1109/TNN.2004.826217.

[95] A. van den Oord and B. Schrauwen, "The student-t mixture as a natural image patch prior with application to image compression", *J Mach Learn Res*, vol. 15, R. Salakhutdinov, Ed., pp. 2061–2086, 2014, ISSN: 1533-7928.

[96] M. Roth, "On the Multivariate t Distribution", 2012.

[97] R. P. Brent, *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

[98] Z Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004, ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861.

[99] R. Franzen. (1999). Kodak Lossless True Color Image Suite, [Online]. Available: http://r0k.us/graphics/kodak/.

[100] K. V. Mardia and P. E. Jupp, *Directional statistics*. John Wiley & Sons, 2009, vol. 494.

[101] P. Kasarapu, "Modelling of directional data using Kent distributions", 2015. arXiv: 1506.08105.

[102] J. T. Kent, "The Fisher-Bingham distribution on the sphere", *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 71–80, 1982.

[103] D. Peel, W. J. Whiten, and G. J. McLachlan, "Fitting Mixtures of Kent Distributions to Aid in Joint Set Identification", *Journal of the American Statistical Association*, vol. 96, no. 453, pp. 56–63, 2001, ISSN: 0162-1459. DOI: 10.1198/016214501750332974.

[104] Y. Rai, P. Le Callet, and P. Guillotel, "Which saliency weighting for omni directional image quality assessment?", in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, IEEE, 2017, pp. 1–6, ISBN: 978-1-5386-4024-1. DOI: 10.1109/QoMEX.2017.7965659.

[105] Y. Rai, J. Gutiérrez, and P. Le Callet, "A Dataset of Head and Eye Movements for 360 Degree Images", in *Proceedings of the 8th ACM on Multimedia Systems Conference - MMSys'17*, New York, New York, USA: ACM Press, 2017, pp. 205–210, ISBN: 9781450350020. DOI: 10.1145/3083187.3083218.

[106] G Strang, *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2016, ISBN: 9780980232776.

[107] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment", in *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, IEEE, pp. 1398–1402, ISBN: 0-7803-8104-1. DOI: 10.1109/ACSSC.2003.1292216.

[108] I. Viola, M. Rerabek, T. Bruylants, P. Schelkens, F. Pereira, and T. Ebrahimi, "Objective and subjective evaluation of light field image compression algorithms", in *2016 Picture Coding Symposium (PCS)*, IEEE, 2016, pp. 1–5, ISBN: 978-1-5090-5966-9. DOI: 10.1109/PCS.2016.7906379.

[109] M. Rerabek and T. Ebrahimi, "New Light Field Image Dataset", in *8th International Conference on Quality of Multimedia Experience (QoMEX)*, 2016.

[110] O. Johannsen, A. Sulc, and B. Goldluecke, "Occlusion-Aware Depth Estimation Using Sparse Light Field Coding", in *Pattern Recognition. GCPR 2016. Lecture Notes in Computer Science*, B Rosenhahn and B Andres, Eds., Springer, Cham, 2016, pp. 207–218, ISBN: 978-3-319-45886-1. DOI: 10.1007/978-3-319-45886-1_17.

[111] Computer Graphics Laboratory. (2008). The (new) Stanford Light Field Archive, [Online]. Available: http://lightfield.stanford.edu/acq.html (visited on 07/19/2019).

[112] O. Johannsen, A. Sulc, and B. Goldluecke, "What Sparse Light Field Coding Reveals about Scene Structure", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016, pp. 3262–3270, ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.355.

[113] Jianqiao Li, Minlong Lu, and Ze-Nian Li, "Continuous Depth Map Reconstruction From Light Fields", *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3257–3265, 2015, ISSN: 1057-7149. DOI: 10.1109/TIP.2015.2440760.

[114] Lytro Inc. (2015). Lytro Desktop Application v5.0.1, [Online]. Available: http://www.lytro.com (visited on 07/10/2017).

[115] T.-C. Wang, J.-Y. Zhu, N. K. Kalantari, A. A. Efros, and R. Ramamoorthi, "Light field video capture using a learning-based hybrid imaging system", *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–13, 2017, ISSN: 07300301. DOI: 10.1145/3072959.3073614.

[116] D. Arthur and S. Vassilvitskii, "K-means++: The Advantages of Careful Seeding", in *SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.

[117] X.-L. Meng and D. van Dyk, "The EM Algorithm-an Old Folk-song Sung to a Fast New Tune", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 59, no. 3, pp. 511–567, 1997, ISSN: 1369-7412. DOI: 10.1111/1467-9868.00082.

[118] P. Liang and D. Klein, "Online EM for unsupervised models", in *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, Association for Computational Linguistics, 2009, pp. 611–619.

[119] M.-a. Sato and S. Ishii, "On-line EM Algorithm for the Normalized Gaussian Network", *Neural Computation*, vol. 12, no. 2, pp. 407–432, 2000, ISSN: 0899-7667. DOI: 10.1162/089976600300015853.

[120] A. W. Moore, "Very fast EM-based mixture model clustering using multiresolution kd-trees", in *Advances in Neural information processing systems*, 1999, pp. 543–549.

[121] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching", in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2000, pp. 169–178.

[122] P. S. Bradley, U. Fayyad, C. Reina, and Others, "Scaling EM (expectation-maximization) clustering to large databases", Technical Report MSR-TR-98-35, Microsoft Research Redmond, Tech. Rep., 1998.

[123] N. Vlassis and A. Likas, "A Greedy EM Algorithm for Gaussian Mixture Learning", *Neural Processing Letters*, vol. 15, no. 1, pp. 77–87, 2002, ISSN: 13704621. DOI: 10.1023/A:1013844811137.

[124] J. J. Verbeek, N. Vlassis, and B. Kröse, "Efficient Greedy Learning of Gaussian Mixture Models", *Neural Computation*, vol. 15, no. 2, pp. 469–485, 2003, ISSN: 0899-7667. DOI: 10.1162/089976603762553004.

[125] N. S.L. P. Kumar, S. Satoor, and I. Buck, "Fast Parallel Expectation Maximization for Gaussian Mixture Models on GPUs Using CUDA", in *2009 11th IEEE International Conference on High Performance Computing and Communications*, IEEE, 2009, pp. 103–109, ISBN: 978-1-4244-4600-1. DOI: 10.1109/HPCC.2009.45.

[126] C. Plant and C. Bohm, "Parallel EM-Clustering: Fast Convergence by Asynchronous Model Updates", in *2010 IEEE International Conference on Data Mining Workshops*, IEEE, 2010, pp. 178–185, ISBN: 978-1-4244-9244-2. DOI: 10.1109/ICDMW.2010.53.

[127] M. C. Altinigneli, C. Plant, and C. Böhm, "Massively parallel expectation maximization using graphics processing units", in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, ACM, New York, New York, USA: ACM Press, 2013, p. 838, ISBN: 9781450321747. DOI: 10.1145/2487575.2487628.

[128] J. Wolfe, A. Haghighi, and D. Klein, "Fully distributed EM for very large datasets", in *Proceedings of the 25th international conference on Machine learning - ICML '08*, ACM, New York, New York, USA: ACM Press, 2008, pp. 1184–1191, ISBN: 9781605582054. DOI: 10.1145/1390156.1390305.

[129] C. Guo, H. Fu, and W. Luk, "A fully-pipelined expectation-maximization engine for Gaussian Mixture Models", in *2012 International Conference on Field-Programmable Technology*, IEEE, IEEE, 2012, pp. 182–189, ISBN: 978-1-4673-2845-6. DOI: 10.1109/FPT.2012.6412132.

[130] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals", *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.

[131] T. Ebrahimi, F. Pereira, P. Schelkens, and S. Foessel, "Grand Challenge on Light Field Coding", Tech. Rep., 2017.

[132] Z. Zhang, C. Chen, J. Sun, and K. Luk Chan, "EM algorithms for Gaussian Mixtures with Split-and-Merge Operation", *Pattern Recognition*, vol. 36, no. 9, pp. 1973–1983, 2003, ISSN: 00313203. DOI: 10.1016/S0031-3203(03)00059-1.

[133] R. Jongebloed, R. Verhack, L. Lange, and T. Sikora, "Hierarchical Learning of Sparse Image Representations Using Steered Mixture-of-Experts", in *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, 2018, pp. 1–6, ISBN: 978-1-5386-4195-8. DOI: 10.1109/ICMEW.2018.8551561.

[134] N. Ahmed, T. Natarajan, and K. Rao, "Discrete Cosine Transform", *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, 1974, ISSN: 0018-9340. DOI: 10.1109/T-C.1974.223784.

[135] V. Avramelos, G. V. Wallendael, and P. Lambert, "Real-Time Low-Complexity Digital Video Stabilization in the Compressed Domain", in *2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, IEEE, 2018, pp. 1–5, ISBN: 978-1-5386-6095-9. DOI: 10.1109/ICCE-Berlin.2018.8576211.

[136] M. Tanimoto, "FTV (free viewpoint television) creating ray-based image engineering", in *IEEE International Conference on Image Processing 2005*, vol. 6, IEEE, 2005, pp. II–25, ISBN: 0-7803-9134-9. DOI: 10.1109/ICIP.2005.1529982.

[137] R. Verhack, L. Lange, P. Lambert, R. Van de Walle, and T. Sikora, "Lossless image compression based on Kernel Least Mean Squares", in *2015 Picture Coding Symposium (PCS)*, IEEE, 2015, pp. 189–193, ISBN: 978-1-4799-7783-3. DOI: 10.1109/PCS.2015.7170073.

[138] D. Sculley, "Web-scale k-means clustering", in *Proceedings of the 19th international conference on World wide web - WWW '10*, New York, New York, USA: ACM Press, 2010, p. 1177, ISBN: 9781605587998. DOI: 10.1145/1772690.1772862.

[139] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description", ITU-T Joint Collaborative Team on Video Coding (JCT-VC), Tech. Rep. JCTVC-S1002, 2014.

[140] ITU-R, "Recommendation ITU-R BT.500-13", International Telecommunication Union, Tech. Rep., 2012, p. 46.

[141] M. Tok, R. Jongebloed, L. Lange, E. Bochinski, and T. Sikora, "An MSE Approach For Training And Coding Steered Mixtures Of Experts", in *2018 Picture Coding Symposium (PCS)*, San Francisco, California USA: IEEE, 2018, pp. 273–277, ISBN: 978-1-5386-4160-6. DOI: 10.1109/PCS.2018.8456250.

[142] E. Bochinski, R. Jongebloed, M. Tok, and T. Sikora, "Regularized Gradient Descent Training of Steered Mixture of Experts for Sparse Image Representation", in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Athens, Greece: IEEE, 2018, pp. 3873–3877, ISBN: 978-1-4799-7061-2. DOI: 10.1109/ICIP.2018.8451823.

[143] R. Jongebloed, E. Bochinski, L. Lange, and T. Sikora, "Quantized and Regularized Optimization for Coding Images Using Steered Mixtures-of-Experts", in *2019 Data Compression Conference (DCC)*, IEEE, 2019, pp. 359–368, ISBN: 978-1-7281-0657-1. DOI: 10.1109/DCC.2019.00044.