UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

**Complexity and Quality Optimization for
Multi-View plus Depth Video Coding**

Tesis Doctoral

Gianluca Cernigliaro

Máster Ingeniero de Telecomunicación

2019

DEPARTAMENTO DE SEÑALES, SISTEMAS Y
RADIOCOMUNICACIONES

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

**Complexity and Quality Optimization for
Multi-View plus Depth Video Coding**

Tesis Doctoral

Autor:

Gianluca Cernigliaro

Máster Ingeniero de Telecomunicación
Politecnico di Torino

Director:

Fernando Jaureguizar Núñez

Doctor Ingeniero de Telecomunicación
Dpto. de Señales, Sistemas y Radiocomunicaciones
Universidad Politécnica de Madrid

TESIS DOCTORAL

**Complexity and Quality Optimization for
Multi-View plus Depth Video Coding**

Autor: Gianluca Cernigliaro

Director: Fernando Jaureguizar Núñez

Tribunal nombrado por el Sr. Rector Magnífico de la Universidad Politécnica de Madrid, el día de de 2019.

Presidente: D.

Vocal: D.

Vocal: D.

Vocal: D.

Secretario: D.

Realizado el acto de defensa y lectura de la Tesis el día de de 2019 en

Calificación:

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

To my family

Resumen

El vídeo 3D, la televisión con elección libre del punto de vista y otros sistemas de vídeo tridimensional, han representado durante años, y todavía representan, una tendencia emergente dentro de las tecnologías de vídeo digital. Una de las representaciones más típicas de vídeo en 3D es el formato Multivista con Profundidad (Multiview plus Depth –MVD). Una escena representada en MVD se captura desde varias cámaras (puntos de vista), capturando diferentes representaciones de la escena desde una gran cantidad de direcciones. Para cada punto de vista se obtiene dos tipos de información: la textura de la escena, representada como una secuencia de vídeo 2D tradicional, con sus componentes de color habituales (RGB o similar), y la geometría de la escena, representada como una secuencia de vídeo en niveles de gris, llamada mapa de profundidad, que contiene la información relacionada con la distancia de los objetos a la cámara. Gracias a las múltiples representaciones de textura más profundidad, una escena 3D puede reconstruirse completamente, proporcionando al usuario la percepción de inmersión en la misma.

Dado que la etapa de compresión es uno de los pasos más importantes en la representación digital de vídeo, la necesidad de codificar eficientemente la información aumenta cuando esta es utilizada para representar la escena en los sistemas 3D. Teniendo en cuenta que un escenario MVD involucra una cantidad creciente de datos debido a los múltiples puntos de vista, y que además cada uno de ellos incluye la nueva información de profundidad, las técnicas de codificación han tenido que evolucionar para minimizar el impacto del creciente volumen de datos y para adaptarse a las características de la información de profundidad.

El trabajo presentado en esta tesis se centra en la adaptación de los métodos tradicionales de compresión basados en AVC/H.264 al entorno MVD. El objetivo perseguido es reducir la carga computacional, que se incrementa dramáticamente por la gran cantidad de representaciones de vídeo, pero también se busca aumentar la eficiencia del proceso de codificación en términos de tasa-distorsión, centrándose en la calidad del vídeo 3D renderizado a través de las múltiples representaciones de color más profundidad. La primera área de investigación ha sido la reducción de la carga computacional de la etapa de Decisión del Modo (Mode Decision –MD), que es una de las de mayor carga computacional del proceso de codificación. La información de geometría proporcionada por los mapas de profundidad ha sido explotada y utilizada para predecir la geometría y el movimiento de los objetos en la escena. Por otro lado, se ha realizado un análisis de la información de profundidad para tener un conocimiento sobre el movimiento en la escena, y que ha proporcionado el entendimiento de cómo está correlacionada la información de movimiento de la componente de textura y de la de profundidad. A continuación, el trabajo se centró en la reducción de la carga computacional de la codificación de los mapas de profundidad usando la etapa de Estimación del Movimiento (Motion Estimation –ME) además de la de MD, y explotando la correlación

existente entre el movimiento de la textura y el de la profundidad. Como resultado, la carga computacional se ha reducido considerablemente en el proceso de compresión con una pérdida de calidad despreciable en la mayoría de los casos. En comparación con la búsqueda exhaustiva de modos y de vectores de movimiento de un codificador AVC/H.264 tradicional, el tiempo consumido se reduce hasta un 40 % en la compresión de la textura y hasta un 58 % en la compresión de la profundidad.

Sin embargo, la reducción de la carga computacional no ha sido el único objetivo del trabajo presentado en esta tesis. Se ha explorado un área considerablemente novedosa, introduciendo nuevos paradigmas de codificación perceptual para la compresión de la profundidad. La última parte de esta tesis se ha centrado en la aplicación de metodologías de percepción, ampliamente explotadas en las técnicas tradicionales de compresión de vídeo 2D, para la compresión de la profundidad. La profundidad se usa solo para fines de reconstrucción 3D como en el caso de la generación de vistas sintéticas. Como esta información nunca se muestra al usuario, los artefactos debidos a su compresión afectarán solo a las representaciones reconstruidas en las vistas sintéticas de la textura. El trabajo de percepción mostrado en esta tesis se ha centrado en adaptar las técnicas tradicionales de compresión perceptiva 2D al formato de representación MVD, optimizando la calidad perceptiva de las vistas sintéticas. El rendimiento de las técnicas perceptivas propuestas para la compresión de profundidad se ha evaluado utilizando métricas de calidad perceptiva, obteniendo una reducción de la tasa de bits de hasta el 13 % con una mejora de hasta 0,3 dB según las mediciones de Bjontegaard.

Abstract

3D Video, Free Viewpoint TV and other three-dimensional imaging systems have represented, and still represent, the emerging trend for digital video technologies. Multi View plus Depth (MVD) is one of the most typical 3D video representations. An MVD scene is recorded from several viewpoints, capturing many different representations from a wide amount of directions. For each viewpoint, two video components are captured: the scene texture, represented as a traditional 2D video with the usual color components (RGB or similar), and the scene geometry, represented as a gray-level image, called depth map, containing the information related to the distance of the scene objects from the viewpoint. Thanks to the multiple texture and depth representations, a 3D scene can be fully reconstructed, providing to the user the perception of immersion.

As for the previous imaging technologies, given that the compression is one of the most important steps of a digital video representation pipeline, also in 3D video has risen the need of encoding efficiently the information used to represent the scene. Considering that an MVD scenario involves an increasing amount of data due to the multiple viewpoints, and also includes new information like the depth maps, the encoding techniques have evolved in order to minimize the impact of the data increasing and to adapt to the depth characteristics. The work presented in this thesis focuses on adapting the traditional compression methods based on AVC/H.264 to the MVD environment, aiming to reduce the computational load, dramatically increased by the high amount of video representations, but also to increase the efficiency of the encoding process in terms of rate-distortion, focusing on the quality of the 3D video rendered through the multiple texture and depth representations.

The first area of research has been the reduction of the computational load of the Mode Decision (MD) stage, which is one of the most computationally expensive of the encoding process. The geometry information provided by the depth maps has been exploited and used to predict geometry and motion of the objects in the scene. On the other hand, analyzing the depth in order to have a knowledge about the motion of the scene has provided an understanding of how the motion information of texture and depth components are correlated to each other. Then, the work has focused on the reduction of the computational load of the depth maps compression, this time involving both MD and Motion Estimation (ME), exploiting the correlation between the motion of the texture and of the depth. The computational load has been considerably reduced in the compression process of both texture and depth maps, reaching up to 40% of reduction in time consumption in the compression of the texture, and up to 58% of reduction in the compression of the depth, when compared to the full search of modes and motion vector of a traditional AVC/H.264 encoder. In both cases, the quality loss has been negligible.

However, the computational load reduction has not been the only goal of the work

presented in this thesis. A considerably novel area has been explored, introducing new perceptual encoding paradigms for the compression of the depth. The last part of this thesis focuses on the application of perceptual methodologies, widely exploited in traditional 2D video compression techniques, but for the compression of the depth. The depth is used only for 3D reconstruction purposes as the generation of the synthetic views, and as it is never shown to the audience, the compression artifacts would affect only the reconstructed representations. The perceptual work shown in this thesis has then focused on adapting traditional 2D perceptual compression techniques to the MVD representation, optimizing the perceptual quality of the synthetic views. The performance of the proposed perceptual techniques applied to depth compression has been evaluated using perceptual quality metrics, reaching a reduction of the bit-rate up to 13% with an improvement of up to 0.3 dB according to the Bjontegaard measurements.

Acknowledgments

There's actually a list people and life events that made this happening... and by this happening I mean me ending a PhD in the video technology field, and taking the time that it took!

The first life event that I can remember is when Dad brought home our first handy camera, one of those with tape, I can't even remember which standard, and I was basically the one playing constantly with it (I also made a real movie with all the kids of my parents' group of friends). So, Dad, thanks for that!

When I started college I was drifting from an interest to another and for couple of years didn't really find passion for any specific college topic. But then, when the video compression was showed to me, it kind of put together, as a weird connection, my love for video and my interest for technology. So I decided to go on focusing on that, choosing all the subjects about image and video I could, and that brought me to meet Fernando, who asked me (I still wonder why) to work in his research group. So, thanks Fernando for being probably drunk that day.

When I was a kid, I was spending a lot of time with my Mum, who's an artist, drawing and painting. Playing with video, at a certain point, took the place of drawing (that, even if just as a kid, I was probably doing much better than researching) but somehow the memories of so many paintings, and art books, and talking about art pieces (as she's also an art teacher) were always there. So at a certain point I decided to make something productive out of all this passions and, as some of you know, I started actually producing video. Creating some horrible stuff initially, but I guess it was just a matter of time and now I must say I'm proud of some results (numbers and comments say it's not bad). That really took a lot of my time and contributed, together to my jobs (money, you know that thing?), to delay this moment until now. So thanks Mum for marking my life since the very beginning, making me interested into too many things and helping me to nearly never end this!

Also to all the people that supported me in my crazy side projects but also to those who, telling me that I was wasting my time, made me work even harder to prove them wrong. Thanks to all of you for contributing pushing away the "responsibility" of ending this!

And finally thanks to those people that, every now and then, were asking "Hey, but weren't you doing a PhD??", for making me feel guilty and making me think about getting back on it. So I did... Here we are! Happy?

Contents

1	Introduction	1
1.1	Binocular vision and stereo visualization	1
1.2	From 2D to 3D	2
1.2.1	Three-dimensional displays	2
1.2.2	3D video applications	3
1.2.3	3D Video compression	5
1.3	Contributions of this thesis	8
1.3.1	Contribution to the texture encoding complexity reduction . . .	9
1.3.2	Contribution to the depth encoding complexity reduction	9
1.3.3	Contribution to the perceptual encoding of the depth sequences	10
2	Mode Decision and Motion Estimation in AVC/H.264 - A brief Introduction	11
2.1	Introduction	11
2.2	Mode Decision	12
2.2.1	Intra Mode Decision	12
2.2.2	Inter Mode Decision	18
2.3	Motion Estimation	21
2.3.1	Motion Estimation in the Inter Prediction	24
2.3.2	Motion Vectors	25
2.3.3	Motion Vectors prediction	28
2.3.4	Inter Skip Mode	28
2.4	Bidirectional prediction	29
2.5	Conclusions	29
3	The Human Visual System and the Perceptual Video Coding - A brief Introduction	31
3.1	Introduction	31
3.2	The Human Visual System	32
3.3	HVS modeling fundamentals	32
3.4	Just Noticeable Difference (JND)	34
3.4.1	JND with pixels	34
3.5	Conclusions	37
4	Fast Mode Decision for Multiview Video Coding based on scene geometry	39
4.1	Introduction	39
4.2	Related work	41
4.3	Depth based Fast Mode Decision (DFMD)	42
4.3.1	Threshold computation	44
4.3.2	Depth Map based Mode Decision	45
4.3.3	Analysis of the surrounding area	46

4.3.4	Conclusions about the DFMD	47
4.4	Disparity and Depth based Fast Mode Decision (DDFMD)	48
4.4.1	DDFMD threshold computation	49
4.5	Complexity reduction analysis	49
4.6	Experimental results	52
4.7	Conclusions	62
5	Low Complexity Mode Decision and Motion Estimation for Depth encoding	63
5.1	Introduction	63
5.2	Related Work	65
5.3	MVD features analysis	68
5.3.1	Sequence features involved in MD and ME	68
5.3.2	The variance of MVD sequences	69
5.4	Low Complexity MD and ME	76
5.4.1	The Depth Moving Edge Detector (DMED)	78
5.5	Complexity reduction analysis	79
5.5.1	Traditional AVC/H.264 computational burden	79
5.5.2	LCMDME computational burden	80
5.6	Experimental results	80
5.6.1	Experimental setup	81
5.6.2	Performance evaluation	83
5.7	Conclusions	95
6	Depth Video Coding for Free Viewpoint Video Oriented to the Synthetic View Perceptual Quality	97
6.1	Introduction	97
6.2	Adopted JND model	99
6.3	Depth Perceptual Encoder (DPE) design	99
6.3.1	Experimental results	101
6.4	Synthetic View Pixel Displacement based Depth Encoder (SVPD-DE) .	105
6.5	Perceptual compression of depth maps based on Just Noticeable Pixel Displacement	107
6.5.1	JND adaptation to depth sequences	107
6.5.2	Just Noticeable Pixel Displacement based Depth Perceptual Encoder (JNPD-DPE)	108
6.6	Experimental results for the SVPD-DE and JNPD-DPE methods	109
6.6.1	Experimental setup	109
6.6.2	RD performance evaluation	110
6.7	Conclusions	113
7	Conclusions and future work	115
7.1	The Good	115
7.1.1	MVD encoding complexity reduction	115
7.1.2	Perceptual Depth Coding	116
7.2	The Bad	116
7.2.1	MVD encoding complexity reduction	116
7.2.2	Perceptual Depth Coding	117

7.3	Future work	117
7.3.1	MVD encoding complexity reduction	117
7.3.2	Perceptual Depth Coding	118
7.4	Some extra considerations	118
7.5	Conclusions	119

List of Figures

1.1	Stereoscope prototype as described in the essay “On some remarkable, and hitherto unobserved, Phenomena of the Binocular Vision” [1]. . . .	2
1.2	The two lenses filter the two projected images allowing only one image to enter each eye.	3
1.3	Parallax-barrier and lenticular autostereoscopic displays.	4
1.4	Scene recorded by a multi-camera system.	4
1.5	A video sequence texture frame (a) with its associated depth frame (b).	6
1.6	Inter-view prediction structure [12].	8
2.1	Example of neighbouring pixels considered for the Intra prediction. . .	13
2.2	Schematic description of the 16×16 Intra Modes.	14
2.3	Example of a 16×16 MB with the corresponding neighbouring area used for the Intra prediction.	14
2.4	Visual representation of the Prediction provided by the four 16×16 Intra Modes applied on the 16×16 MB shown in Fig.2.3, and the corresponding Sum of Absolute Errors (SAE).	15
2.5	Example of neighbouring pixels considered for the 4×4 Intra Modes. . .	16
2.6	Graphical representation of the 4×4 Intra prediction Modes.	17
2.7	Visual representation of the Prediction provided by the nine 4×4 Intra Modes applied on a 4×4 block used as example.	18
2.8	Example of the searching area of the previous frame considered for the Inter prediction.	19
2.9	Partitions of a 16×16 MB: 16×16 , 8×16 , 16×8 , 8×8	20
2.10	Partitions of an 8×8 sub-block: 8×8 , 4×8 , 8×4 , 4×4	21
2.11	Residual Image of the difference between Frame 1 and Frame 2.	22
2.12	Residual image of the difference between two frames with the corresponding Inter modes selected for the compression.	23
2.13	Example of search of the best matching on the reference frames for one MB of the current frame to be encoded.	23
2.14	Four MBs area of a frame to encode with 4 different Inter prediction modes.	24
2.15	Example of the best matching area found for two 8×16 sub-blocks. . .	25
2.16	Example of a MV with magnitude $(1, -1)$	26
2.17	Example of a MV with magnitude $(0.75, -0.5)$	26
2.18	Interpolation of luminance half-pel positions.	27
2.19	Interpolation of luminance quarter-pel positions.	27
2.20	A, B and C MBs MVs used for the prediction of the MV of the MB E.	28
2.21	A, B and C sub-blocks MVs used for the prediction of the MV of the MB E.	29

2.22	Three different examples of Bidirectional prediction: (a) past/future, (b) past, (c) future.	30
3.1	Human eye cross section.	32
3.2	Stages of the HVS modeling.	33
3.3	Visibility threshold $T^L(i, j)$ in function of the average background luminance $L(i, j)$	35
3.4	Temporal masking effect function $f(d)$ in function of the average inter-frame difference $d(i, j)$	36
4.1	Overall architecture of the DFMD algorithm.	43
4.2	Flow diagram describing all the stages of the threshold evaluation. . .	44
4.3	Example of the overlapping between the RV selected mode and the depth MB for the CV.	44
4.4	Example of the depth maximum values in a MB with three 8×8 partitions and one 4×4 submode.	45
4.5	Flow diagram describing all the stages of the Depth based MB partition for the CV.	46
4.6	MBs surrounding the current one (shaded) for the 16×16 analysis. . .	47
4.7	Flow diagram of the DFMD Algorithm	48
4.8	Bad correspondence between RV and CV in case of disparity.	49
4.9	Depth map's segmentation using disparity.	50
4.10	Sequence Akko&Kayo. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.	56
4.11	Sequence Mobile. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.	56
4.12	Sequence Balloons. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.	57
4.13	Sequence Beergarden. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.	57
4.14	Sequence Book Arrival. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.	58
4.15	Sequence Cafe. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.	58
4.16	Sequence Akko&Kayo. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.	59

4.17	Sequence Mobile. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.	59
4.18	Sequence Balloons. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.	60
4.19	Sequence Beergarden. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.	60
4.20	Sequence Book Arrival. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.	61
4.21	Sequence Cafe. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.	61
5.1	Encoding time comparison between texture and depth for a set of 5 MVD sequences (<i>Beergarden</i> , <i>Book Arrival</i> , <i>Cafe</i> , <i>Kendo</i> and <i>Newspaper</i>) . .	66
5.2	Frame #2 of the 5th view of <i>Beergarden</i> (a) with its associated depth map (b).	70
5.3	MB-wise temporal and spatial variance for the texture and depth frame #2 of the 5th view of the sequence <i>Beergarden</i>	72
5.4	Comparison between texture and depth spatial MB-wise variances evaluated on a set of five MVD sequences (<i>Beergarden</i> , <i>Book Arrival</i> , <i>Cafe</i> , <i>Kendo</i> , <i>Newspaper</i>)	73
5.5	Comparison between texture and depth temporal MB-wise variances evaluated on a set of five MVD sequences (<i>Beergarden</i> , <i>Book Arrival</i> , <i>Cafe</i> , <i>Kendo</i> , <i>Newspaper</i>)	75
5.6	Flowchart of the proposed LCMDME algorithm.	77
5.7	Sequence <i>Beergarden</i> . RD and Depth encoding time comparison of the LCMDME vs Full Search ME, EPZS ME and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	84
5.8	Sequence Book Arrival. Comparison of the PSNR performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	86
5.9	Sequence Cafe. Comparison of the PSNR performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	86
5.10	Sequence Kendo. Comparison of the PSNR performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	87

5.11	Sequence Newspaper. Comparison of the PSNR performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	87
5.12	Sequence Mobile. Comparison of the PSNR performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	88
5.13	Sequence Pantomime. Comparison of the PSNR performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	88
5.14	Sequence Book Arrival. Comparison of the VQM performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	89
5.15	Sequence Cafe. Comparison of the VQM performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	89
5.16	Sequence Kendo. Comparison of the VQM performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	90
5.17	Sequence Newspaper. Comparison of the VQM performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	90
5.18	Sequence Mobile. Comparison of the VQM performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	91
5.19	Sequence Pantomime. Comparison of the VQM performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	91
5.20	Sequence Book Arrival. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	92
5.21	Sequence Cafe. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	92
5.22	Sequence Kendo. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	93
5.23	Sequence Newspaper. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	93
5.24	Sequence Mobile. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> (<i>th.</i> = 3, 10).	94

5.25	Sequence Pantomime. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu <i>et al.</i> ($th. = 3, 10$).	94
6.1	Comparison of the perceptual performance of the DPE vs. a traditional AVC/H.264 encoder for the sequences <i>Mobile</i> , <i>Beergarden</i> and <i>Kendo</i>	104
6.2	VQM comparison between the sequences encoded through the proposed algorithms and the traditional AVC/H.264 for the sequence <i>Balloons</i>	111
6.3	VQM comparison between the sequences encoded through the proposed algorithms and the traditional AVC/H.264 for the sequence <i>Mobile</i>	111
6.4	PSNR comparison between the sequences encoded through the proposed algorithms and the traditional AVC/H.264 for the sequence <i>Balloons</i>	112
6.5	PSNR comparison between the sequences encoded through the proposed algorithms and the traditional AVC/H.264 for the sequence <i>Mobile</i>	112

List of Tables

4.1	Computational complexity reduction.	51
4.2	MVD sequences used for the experiments.	52
5.1	Pearson correlation coefficients evaluated on the temporal and spatial variances between texture and depth.	74
5.2	Pearson correlation coefficients evaluated on the temporal and spatial variances between texture and depth for every second of the sequence <i>Cafe</i>	76
5.3	Computational complexity burden required for the MD/ME algorithm.	80
5.4	MVD sequences used for the experiments.	81
5.5	Bit rate and PSNR difference with the Bjontegaard measure between the Full Search based AVC/H.264 and the compared four methods. . .	85
5.6	Average saved depth encoding time of the LCMDME method with re- spect to the Full Search ME.	85
5.7	Percentage of Full Search MB for the limit cases (<i>Beergarden</i> and <i>Book Arrival</i>).	95
6.1	MVD sequences and settings used for the experiments.	102
6.2	MVD sequences and settings used for the experiments.	110
6.3	Bit rate and PSNR difference between the traditional AVC/H.264 and the proposed methods.	113

List of Acronyms

3DTV	Three-Dimensional Television
3DV	Three-Dimensional Video
AVC/H.264	Advanced Video Coding
CV	Current View
CVQM	Command line Video Quality Metric
DCT	Discrete Cosine Transform
DDFMD	Disparity and Depth based Fast Mode Decision
DFMD	Depth based Fast Mode Decision
DIBR	Depth-Image-Based Rendering
DMED	Depth Moving Edge Detector
DPE	Depth Perceptual Encoder
DV	Disparity Vector
EPZS	Enhanced Predictive Zonal Search
FVV	Free Viewpoint Video
GOP	Group of Pictures
HC	High Complexity
HVS	Human Visual System
JM	Joint test Model (Reference Software for AVC/H.264)
JND	Just Noticeable Difference
JNPD-DPE	Just Noticeable Pixel Displacement based Depth Perceptual Encoder
ICIP	International Conference on Image Processing
LC	Low Complexity
LCMDME	Low Complexity Mode Decision and Motion Estimation
MB	Macro Block
MD	Mode Decision
ME	Motion Estimation
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
MSPE	Mean Squared Perceptual Error
MV	Motion Vector
MVD	Multi View plus Depth
PCC	Point Cloud Compression

PCS	Picture Coding Symposium
PD	Perceptual Distortion
PSNR	Peak Signal to Noise Ratio
PSPNR	Peak Signal to Perceptual Noise Ratio
PST	Pixel Shift Tolerance
QP	Quantization parameter
RD	Rate-Distortion
RDO	Rate-Distortion Optimization
RS	Reference Software
RV	Reference View
SAD	Sum of Absolute Differences
SAE	Sum of Absolute Errors
SJND	Spatial Just Noticeable Difference
SSD	Sum of Squared Difference
SVPD-DE	Synthetic View Pixel Displacement based Depth Encoder
VQM	Video Quality Metric

1 Introduction

Chapter Abstract

The goal of this thesis is the presentation of several novel approaches for the optimization of the Video plus Depth compression process. This Introductory chapter has the purpose to provide to the reader an overview of the 3D video environment where the proposed work has been developed, giving a brief introduction of the 3D vision systems and of the main characteristics of the 3D visualization techniques, as the 3D display and the related applications. This chapter ends with a description of the research work presented in this thesis, explaining the main areas of focus, the novelty of the designed methods and, finally, enumerating the papers presented in international conferences, symposiums and in journals, where the research has been published.

1.1. Binocular vision and stereo visualization

The human eye and brain are designed to visualize and represent the world around us. The human vision system is able to process the geometry of the surrounding scene elaborating and reconstructing the three dimensions in the inner human brain. Thanks to this process, we have several abilities, as the perception of the position of the objects, and the distance of them from us. The human brain is able to process all these different types of information, creating a reconstruction of the observed 3D world. This happens exploiting the signals sent by the eyes and processed by the human binocular vision system.

The human binocular vision has been subject of studies since 1838, when Charles Wheatstone wrote his essay “On some remarkable, and hitherto unobserved, Phenomena of the Binocular Vision” [1]. Wheatstone studied the laws of perspective, developing a study based on the so obvious idea that the visual perception of an object changes according to its position. However, the visual perception also changes depending on which eye we use to see it, in fact, if an object is placed in front of us, so near that, to focus on it, we must converge the eyes, each eye will see a different image of the same object. On the other hand, when objects are placed at great distances, both eyes see almost the same image. The three dimensions of the scene are perceived by the brain thanks to this double projection, because to focus something at a specific distance we need to move the eyes at a specific degree of convergence. Such convergence tells if that object is closer than another one.

Wheatstone was the first studios who started reproducing experimentally an artificial perception of a three-dimensional space, projecting two images reproducing the same object from two different perspectives. According to him, sending to the retina of

each eye an image of the same object, represented from a different point of view, the perception of the third dimension can be simulated in the human brain. To proof his theory, he adduced an experiment made on the “stereoscope” (Figure 1.1), which is a prototype of a rudimentary stereoscopic display, where two handmade drawings (E' and E), representing two cube faces viewed from two different perspectives, were projected to the retinas by two mirrors placed in front of them (A' and A). Thanks to the stereoscope, the brain was able to recreate the 3D perception from two 2D images.

Figure 1.1: Stereoscope prototype as described in the essay “On some remarkable, and hitherto unobserved, Phenomena of the Binocular Vision” [1].

1.2. From 2D to 3D

Nowadays, the technology improved the three dimensional experience developing high fidelity 3D reconstructions. As a consequence, video applications that were already used in traditional 2D video representation have been adapted and renewed, embracing this more immersive video representation. This section of the introduction shows how these applications have changed, providing an overview of three main areas. First, the three-dimensional displays, used for the visualization of the 3D images and video; second the rendering applications born to support the 3D technologies, as Free Viewpoint Video (FVV) or three-dimensional Television (3DTV); finally the 3D video compression techniques, which represent the core field of study around which the work described in this thesis has been developed.

1.2.1. Three-dimensional displays

The stereoscope experiment is the ancestor of all the three-dimensional stereoscopic displays that, nowadays, allow the observer to watch a 3D scene looking at two bi-dimensional images. The first stereoscopic monitors were indeed designed to reproduce two overlapped images (or views) showing the same video from two different perspectives (left and right images) as if they were seen from two different eyes. Those two

images were both reproduced on the same screen at the same time (or one after the other) and, thanks to additional devices as special glasses, every image was projected to one eye and filtered for the other one, recreating the three-dimensional perception as in the experiment of Wheatstone (Figure 1.2).

Figure 1.2: The two lenses filter the two projected images allowing only one image to enter each eye.

The goal of these three-dimensional technologies is clearly the improvement of the immersive experience, enhancing the fidelity and the realism of the world representation. Today, thanks to the new technological advances, the stereoscopic monitors are not the only way to show 3D images. Autostereoscopy has been introduced among the 3D displays giving the chance to observe 3D images without stereoscopic glasses or other additional devices [2].

Autostereoscopic displays mainly employ two technologies called, respectively, parallax barriers and lenticular lenses. Both technologies are able to redirect, to each eye, a different image according to the position of the observer. The first one (parallax barriers) is a device placed in front of an image source, able to hide one or more views to one eye of the observer, redirecting to each eye only the view needed to create the 3D perception. The second (lenticular lenses) adopts the same principle, showing to each eye only one view, but in front of the images there is an array of lenses, designed to redirect different images to different angles. Figure 1.3 shows a graphic representation of how both displays redirect different images to different angles. A disadvantage of both technologies is that the viewer must be placed in a well-defined spot to experience the 3D effect.

1.2.2. 3D video applications

Together with the 3D video displays, the technology has evolved in order to adapt to the characteristics of the new video signal and to exploit new possibilities. FVV

Figure 1.3: Parallax-barrier and lenticular autostereoscopic displays.

and 3DTV [3], [4], represent an example of new applications created around the three-dimensional video. In both cases the scene is captured by a multi-camera system, where a set of N cameras is placed around the subject (Figure 1.4) recording N video sequences called views. Every view reproduces the same scene and the same objects from a different point of view. Thanks to this multiple representation, it is possible to handle various different perspectives of the same scene and redirect a different view to each eye through the 3D displays, as explained in Section 1.2.1. Redirecting to each eye the scene from the corresponding perspective, it is possible to give an immersive sensation to the observer.

Figure 1.4: Scene recorded by a multi-camera system.

Immersion performance depends on the number of viewpoints that, if high, improves the feeling of the 3D experience. However, increasing the number of cameras capturing

the scene, a considerable amount of information needs to be transmitted or recorded [5]. This translates into a higher load of data to process, higher storage space and/or bigger bandwidth needed. To overcome these disadvantages the number of viewpoints captured should be limited. However, as a consequence, this limitation means a worse 3D perception. The solution to this issue is finding the best trade-off between the number of viewpoints and the perception of the reality. In addition, the point of view from a position where a camera is not physically placed can be generated synthetically [6].

The generation of the synthetic, or virtual, views is the basis of FVV. The virtual view generation allows the observer to watch a video sequences recorded from a position where a real camera is not physically placed. Therefore, the observer has the chance to watch a scene having a complete freedom in the selection of the position of the point of view. However, the virtual view generation needs to be set in a well-defined environment where one or more video sequences are recorded from real cameras with an additional information that provides the knowledge about the distance, from the camera, of the objects represented in the scene. This information, called depth information (or depth maps/sequences), consists of gray level video sequences showing the depth of the objects presented in the same scene captured by the traditional cameras. Depending on the distance of the objects, the depth is represented as a set of values that range from black (0), corresponding to the deepest background, to white (255), corresponding to the nearest objects [7]. This setting of traditional video sequence with the additional depth information is called View plus Depth and, when several views are involved in the capture, is considered a Multi View plus Depth (MVD) environment [8].

The traditional video sequences representing the color are also called textures. Figure 1.5 shows an example of a texture frame with the corresponding depth frame.

In this environment, the virtual views are generated by an interpolation of the reference textures, captured by real cameras, warped to the location of the virtual one, using the depth information to locate the objects in the 3D space. One of the most used synthetic view generation methods is called Depth-Image-Based Rendering (DIBR) [9]. In the DIBR algorithm, knowing the depth information and the cameras parameters, it is possible to generate a synthetic view by projecting the real view pixels to the positions that they would have if they were seen from the viewpoint placed where the virtual camera has to be generated.

1.2.3. 3D Video compression

Video compression advances are usually developed in parallel to the new video technologies. The reason is that new characteristics, like higher resolution or increasing amount of cameras, request the adaptation of the encoding strategies to exploit new encoding paradigms to optimise the compression. The consequence of releasing new three-dimensional video applications has been that the video compression techniques needed a renovation, oriented to the 3D video representation.

The presence of the multi-camera environment, as expected, produced new issues regarding the larger amount of data to handle and because of the presence of new typolo-

(a) Texture Frame

(b) Depth Frame

Figure 1.5: A video sequence texture frame (a) with its associated depth frame (b).

gies of information as the depth. Due to the different video characteristics and due to the presence of the 3D video which is no longer composed by a single video sequence, the development of new encoding scenarios has become a necessity.

As explained in Section 1.2.2, the 3D scene is recorded by a large amount of cameras, generating information that is represented as texture plus depth sequences. As a consequence of the presence of a high number of sequences and of the depth information, the research about 3D video compression needs to solve a twofold problem. First, the high number of cameras increases the number of video sequences to compress, so also the information to manage is considerably increased. This increase of data also means a higher computational burden, so the encoders need longer processing time or better calculating capacity. Second, the depth information is represented through additional video sequences which, instead of the traditional ones, have very specific characteristics as absence of colors, textures and very different properties related to the frequency. Due to these differences, the depth data have to be appropriately managed through specific techniques.

Analyzing the new characteristics of the 3D video environment, new compression strategies based on different approaches have been considered. Strategies able to solve, or at least reduce, the problems given by the increased amount of information to process and the issue of the different characteristics of the depth sequences. An example of the adaptation of the video compression techniques towards the 3D is the Multiview Video Coding standard [10], an evolution from AVC/H.264 toward a multi-camera compression system that, in addition to the temporal one, reduces also the inter-camera correlation.

Different views recorded by cameras placed close to each other capture the same scene from slightly different points of view. The frames of these sequences will have, then, a high inter-camera redundancy. The commonly used temporal prediction techniques can be then "translated" to an inter-frame spatial prediction: a frame recorded from one view can be used as a reference for the encoding of the frame recorded at the same time from a second view captured by a camera placed in a very close position. Thanks to this strategy, it is possible to further reduce the signal redundancy and, as consequence, increase the compression ratio. New prediction schemes have been studied by Smolic *et al.* [11] and by Merkle *et al.* [12] who proposed new structures for the prediction. Figure 1.6 shows an example of a new prediction structure used to optimize the compression in the encoding of a 3D scene recorded by a multi-camera system.

This inter-view prediction scheme does not solve the problems of i) the high computational burden and ii) the presence of the depth information which increases the amount of information to be compressed that, differing from the traditional video in terms of frequency of the content and in terms of color representation, may need different approaches. The research work, developed in this thesis, focuses mainly on these two issues. In the following section, an overview of the contribution of this thesis is given (Section 1.3).

Figure 1.6: Inter-view prediction structure [12].

1.3. Contributions of this thesis

This thesis presents the research, and the related results, of several proposed strategies for the encoding of 3D video sequences captured in an MVD environment where, as mentioned before, the video information is represented by a high number of traditional 2D videos, used for the texture/color components, and the corresponding depth maps.

This kind of information has created several issues as the high amount of data to process (and the corresponding increase of computational load) or the presence of the depth maps that, being represented as gray-scale images, have a quite different response to traditional video coding techniques.

Together with the issues to solve, some new challenges have raised, as the inter-view correlation to be exploited, the similarities between depth and textures, or the fact that, in an MVD environment, the viewpoints are used to generate new content as the synthetic views, to give view freedom to the observer.

The work described in this thesis aims to solve in particular the challenge of reducing the computational burden by exploiting the aforementioned correlations and similarities that have been arising, but also to focus on a new way to face the compression of the depth maps, based on the perceptual quality of the final product that is delivered to the user: the synthetic views. The work covers then the following two main topics:

- Computational burden reduction of the MVD encoding process.
- Perceptual quality improvement of the compressed MVD sequences.

The thesis has been divided into three main areas. First, the computational burden reduction of an MVD environment is considered, focusing on the complexity reduction of the texture signal encoding. Second, it continues with the computational burden reduction, focusing exclusively on the encoding process of the depth content. Finally,

the last part, focuses in detail on the depth maps encoding, improving the encoding efficiency considering the perceptual quality of the 3D representation.

1.3.1. Contribution to the texture encoding complexity reduction

As previously stated, in an MVD environment the scene is represented by texture and depth sequences (Section 1.2.2). The depth sequences are used at the display stage to create virtual views where a camera is not physically present.

The depth information is an auxiliary information used to provide the distance of the objects from the camera. However, in addition, also provides a different description of the same scene represented by the texture. In a depth sequence the separation between the objects shown in the scene, when they are in different depth levels, is indeed much more defined than in the texture. This consideration can be done as well about the motion of the scene that, in some cases, can be defined better at a depth level. Exploiting the depth, then, it is possible to reduce the number of operation needed to process one of the heaviest stages of the encoding process: the Mode Decision (MD) stage. In the first part of this thesis, a study about the MD complexity reduction that exploits the depth discontinuities has been described.

The study developed has been divided into two parts. First, a preliminary research based on the knowledge of the depth of the scene, has been published in the *SPIE Visual Communication and Image Processing Conference* of 2009 [13]. Second, this study has been further developed introducing the disparity between views as information to exploit. The disparity is the spatial difference between areas placed in different frames recorded at the same time by different cameras. The improved results have been published in the *IEEE International Conference on Image Processing (ICIP)* of 2010 [14].

1.3.2. Contribution to the depth encoding complexity reduction

Depth sequences are videos representing the same scene as the one represented in the texture ones, but with different characteristics (gray scale depending on the distance of the objects from the camera). However, analyzing similarities and differences it is possible to notice that, given that the objects in both representations are the same and have the same motion, in many cases the encoding strategies for the compression of similar areas of a depth sequence, and of the corresponding texture, could be the same. When a texture sequence is already encoded, the reuse of the information which have been previously evaluated makes possible the reduction of the computational burden of two heavy stages of the depth encoding: Mode Decision (MD) and Motion Estimation (ME). The second main topic presented in this thesis shows how it is possible to exploit the motion similarities between texture and depth to reduce the complexity of the depth sequences encoding process.

Also this work has been divided into two parts: the first one has been presented in the *IEEE International Conference on Image Processing (ICIP)* of 2011 [15]. The preliminary results have been further analyzed observing the statistical features of the depth and of the textures obtaining improved performance able to reduce the depth

encoding complexity without losses in terms of quality. These last results have been published in the *IEEE Transaction on Circuits and Systems for Video Technology* [16].

1.3.3. Contribution to the perceptual encoding of the depth sequences

Depth sequences, due to their different purposes, are never shown to the observer. They are indeed only used to know the distance of the objects from the camera, to be able to create virtual views where a camera is not physically placed, generating videos known as synthetic views.

Normally, depth maps are handled as traditional video sequences and compressed using the same techniques designed for traditional video. The main drawback of using traditional encoding strategies on the depth video is that the compression results are optimized according to the reconstructed depth distortion introduced by the encoder. But, if depth sequences are never shown, this is definitely not the best way to compress this kind of information.

In addition, if in the depth encoding process we consider that the specific goal of the presence of the depth content is the generation of synthetic views, the compression efficiency can be improved using other strategies. So, the last part of the thesis presents a new way to compress the depth information, designed considering the perceptual quality of the synthetic views. Considering the distortion of the synthetic view, rather than the one of the decoded depth, the encoding resources can be better exploited and the compression can be processed according to the video quality perceived by the observer.

In this thesis two Depth Perceptual encoding algorithm are presented. The first one shows a study about the possibility of encoding the depth considering the synthetic view perceptual quality and it has been presented at the *Picture Coding Symposium (PCS) of 2012* [17]. The limitations of this first study have been analyzed and an improved algorithm has been developed, able to obtain a better perceptual quality and, in some cases, also improving the compression quality according traditional objective metrics. This last work has been published in the *IEEE International Conference on Image Processing (ICIP)* of 2013 [18].

2 Mode Decision and Motion Estimation in AVC/H.264 - A brief Introduction

Chapter abstract

Mode Decision (MD) and the Motion Estimation (ME) are two crucial stages of the AVC/H.264 encoding process. The reader can consider this section, which explains the two aforementioned stages, as an introduction to the work presented in the Chapters 4 and 5. The main topic of the aforementioned chapters is, indeed, the optimization of MD and ME in the encoding process of MVD video sequences.

2.1. Introduction

The goal of the video coding process is the reduction of the amount of data needed to store or transmit the information representing a digital video signal. The video signal, without any kind of compression, would have such a high bit rate that it would be difficult, or impossible, to be handled, especially in real time environment.

As in other compression areas, video coding algorithms exploit the correlation of the signal in order to identify the information that is redundant. In image compression, for example, the spatial redundancy of the data is exploited: pixel values are predicted considering the neighbouring area and only the difference between the prediction and the real value is encoded. Using this strategy, only high frequency areas need a high volume of data to be described. On the other hand, homogenous areas are compressed saving a considerable amount of information.

Video coding can be considered as an evolution of the ideas behind image compression. The strategies that exploit the spatial redundancy are, indeed, very similar to the one used in image compression. In addition, given the presence of a frame by frame motion, video coding can also exploit the temporal redundancy between two images. In order to reduce the aforementioned redundancy, certain rules have been developed and defined in video compression standards that have been evolving together with the digital video.

The majority of the contemplated rules consider the prediction of the area of a frame the most important stage of the video coding process. The better is the prediction, the easier is the redundancy reduction. The rules explained in this chapter are the ones belonging to the standard AVC/H.264 [19]. They are defined in order to obtain the best decision about which image area can be considered for the prediction of other areas. In other words, some pixels, previously encoded, will be used as reference for

the estimation of the value of the pixels to encode. The considered rules are defined in the standard and are called *prediction modes*.

The prediction mode of a certain area of a frame defines exactly which pixels of the previous, current or following frame, are considered in the encoding. In addition, it defines how the estimation of the pixels to encode is calculated. Depending on the characteristics of the pixels of the area to encode, one prediction mode can provide better compression results than the others. To decide which is the best prediction mode, a stage called Mode Decision (MD) is needed in the encoding process. In the first part of this chapter, all the possible prediction modes defined in the standard AVC/H.264, and used for the MD stage, are described.

As introduced above, given the temporally correlated nature of the video signal, also the temporal redundancy reduction has an important role in the video compression process.

In order to exploit, in the best possible way, the temporal correlation and to maximize the bit stream compression, the difference between frames needs to be evaluated. As, most of the time, the frames considered are consecutive frames, the difference is related to the *motion* of the video signal: the process used in AVC/H.264 to perform such evaluation is indeed called Motion Estimation (ME). The second part of this chapter describes all the steps of the ME stage.

2.2. Mode Decision

In AVC/H.264, the compression process is based on coding units, each basic coding unit is called Macro Block (MB). A MB is a squared region of 16×16 pixels; the pixels belonging to each MB are processed separately from the pixels belonging to other areas. For each MB, a different encoding strategy is considered among a set of possible options. Every option will represent the encoded MB with a specific amount of bits and generating a certain compression distortion from the original video. The MD process is in charge to evaluate, for all the compression strategies, the amount of bits needed to represent a MB and the respective distortion.

In this section, an overview of all the strategies that an AVC/H.264 encoder can consider in the MD process, is provided.

2.2.1. Intra Mode Decision

As previously explained, a MB is encoded considering the previously encoded information and, when it belongs to the same frame, the compression is performed exploiting the spatial correlation. The strategies that exploit the spatial redundancy, within the same frame, are called Intra Modes.

Intra Modes are basically coding modes where the previously encoded information of the same frame is considered to evaluate the prediction of the current MB. The pixels, already processed, that surround the MB to encode, are used to predict the current MB replicating their values or extrapolating new ones. In Figure 2.1 it is possible to observe

an example of a 16×16 pixel MB and the surrounding pixels used for the prediction. Such prediction is done considering the row of pixels marked as H (horizontal pixels) and the column of pixels marked as V (vertical pixels).

Figure 2.1: Example of neighbouring pixels considered for the Intra prediction.

When a MB is encoded considering an Intra Mode, the prediction can be evaluated on the whole area using one of the several possible 16×16 Intra Modes or, separately, considering sixteen independent 4×4 pixel sub-blocks using one of the 4×4 Intra Modes.

16×16 Intra Modes

When the 16×16 Intra Modes are considered for the MB encoding, the prediction is evaluated considering the whole MB and the decided strategy is applied on the 16×16 area. The four possible 16×16 Intra Modes are:

- 0 Vertical Prediction
- 1 Horizontal Prediction
- 2 Mean Prediction
- 3 Plane Prediction

Depending on the mode considered, the pixels of the MB are predicted following different rules. In Figure 2.2 it is possible to observe a graphical description of the 16×16 Intra Modes. In the Vertical and the Horizontal prediction modes, the surrounding pixels are replicated to estimate the values in the MB. In Mode 0 (Vertical) the values of the row H are vertically replicated creating the prediction values for the 16×16 pixels of the MB. On the other hand, in Mode 1 (Horizontal) a similar process is applied replicating the vertical left column (V) horizontally.

In the Mode 2 (Mean Prediction), instead, the pixels are estimated by interpolation. The average value among the pixels of the horizontal upper row (H) and of the vertical left side column (V) is calculated and it is used to predict the value of all the pixels of the 16×16 MB.

Finally, in Mode 3 (Plane Prediction), the horizontal upper row and the vertical left side column are replicated diagonally. When Mode 3 is processed, the value of every second pixel of the row H is replicated diagonally in the MB; the same happens for the vertical column V in the opposite direction.

Figure 2.2: Schematic description of the 16x16 Intra Modes.

The variety of the strategies used in the Intra Modes is designed to face different pixel patterns. The Mean Mode (2), for example, is very suitable for uniform areas because, with one value, the whole MB is predicted with a good precision. The Vertical, Horizontal and Plane Modes (0, 1 and 3) are, instead, able to predict with high accuracy repetitive patterns or straight textures and edges (vertical, horizontal and diagonal).

The AVC/H.264 encoder, usually, tests all the possible strategies and selects the one that minimises the error according to a specific distortion metric.

Figure 2.3: Example of a 16x16 MB with the corresponding neighbouring area used for the Intra prediction.

Figure 2.3 and 2.4 show the graphical application of the four 16x16 Intra Modes. Figure 2.3 shows a 16x16 MB with the corresponding neighbouring area used for the Intra prediction (upper horizontal row and left vertical column), and Figure 2.4 shows the application of all the possible 16x16 Intra Modes and the distortion evaluated between the original MB and the four predicted versions.

Figure 2.4.a shows the application of the Mode 0 (Vertical), where the reference used is the upper horizontal row H. All the pixels of the MB are a replication of the reference row.

Figure 2.4.b shows the application of Mode 1 (Horizontal), where the reference pixels are the ones belonging to the left vertical V column and they are all horizontally replicated on the MB.

Figure 2.4.c shows the Mean Prediction, where all the pixels of the MB are predicted as the average value of the reference pixels.

Finally, Figure 2.4.d shows the diagonally replicated pixels estimated when the Plane Prediction is used.

Below each image, it is represented the Sum of Absolute Errors (SAE) between the predicted MB and the original one (shown in Figure 2.3). The Intra Mode that minimises the SAE (Mode 3, plane) is the most suitable one for the reduction of the redundancy, because allows the encoder to use the lowest possible amount of data to represent the original MB.

Figure 2.4: Visual representation of the Prediction provided by the four 16x16 Intra Modes applied on the 16x16 MB shown in Fig.2.3, and the corresponding Sum of Absolute Errors (SAE).

4×4 Intra Prediction

In certain situations, considering the neighbouring row and column used by the 16×16 Intra Modes to encode a whole 16×16 MB may not be the best solution. If, in a MB, there is a big variety of edges, shapes and patterns, there may be the need of different prediction modes applied in smaller areas of the image (smaller than 16×16 pixels). For that reason, the Intra prediction rules include also the 4×4 Intra Modes.

The principle behind the 4×4 Intra prediction strategies is pretty similar to the 16×16 one. The differences are given by the number of modes, that are nine instead of the four previously explained, and by the number and the position of the pixels of the surrounding area used to generate the candidate blocks.

The surrounding area, used as reference pixels for the prediction, considers also pixels that are not adjacent to the area to encode. The reference pixels used are in fact the four ones belonging to the left column beside the block, plus the nine ones laying above the block starting from the one placed on the $(i - 1, j - 1)$ position, where the pixel placed on the (i, j) position is the one in the top left corner of the 4×4 block to compress. Figure 2.5 shows, in capital letters, which are the previously encoded pixels, surrounding the 4×4 block, that are considered for the prediction in lower case.

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Figure 2.5: Example of neighbouring pixels considered for the 4×4 Intra Modes.

4×4 Intra Modes

The modes considered in the 4×4 Intra prediction use similar rules to the 16×16 ones like replication and extrapolation. The difference is that there are six diagonal replication modes covering more possible directions. These nine possible 4×4 Modes are:

- 0 Vertical Prediction
- 1 Horizontal Prediction
- 2 Mean Prediction
- 3 Diagonal Down-left Prediction
- 4 Diagonal Down-right Prediction
- 5 Vertical Right Prediction
- 6 Horizontal Down Prediction
- 7 Vertical Left Prediction

■ 8 Horizontal Up Prediction

A graphical description of the nine 4×4 Intra Prediction Modes is shown in Figure 2.6. The description of the replication modes shows which pixels, belonging to the ones used as reference, are used to build the predicted block and in which direction they are replicated. It is possible to observe the six different diagonal replication modes (3, 4, 5, 6, 7, 8) that, together with the vertical and horizontal replication (0 and 1), cover a wider number of possibilities than the 16×16 ones. In addition, it is possible to observe that a larger amount of reference pixels, in proportion, is needed to cover this higher variety of replication directions, i.e. pixels E, F, G and H are used to predict some of the 4×4 block pixels in the diagonal down-left prediction (3).

Figure 2.6: Graphical representation of the 4×4 Intra prediction Modes.

In order to better explain the nine 4×4 modes, a specific example is shown in Figure 2.7.

Also in this case, it is possible to observe the generation of the predicted areas, created replicating or interpolating the reference pixels. As in the example shown for the 16×16 Intra Modes, the distortion is evaluated calculating the SAE for every possible strategy. The minimum SAE is obtained using the Mode 8 (Horizontal Up) that minimises the amount of data needed to represent the 4×4 block.

Figure 2.7: Visual representation of the Prediction provided by the nine 4×4 Intra Modes applied on a 4×4 block used as example.

2.2.2. Inter Mode Decision

The previous section analyzed how an AVC/H.264 based video encoder considers the spatial redundancy to compress the video signal (Section 2.2.1) with the Intra Modes. This current section shows how the temporal redundancy is exploited, defining the rules used to temporally predict every single frame and, finally, which is the information used to represent the temporal based compression.

Reference Frame and Search Area

In the Intra Mode Decision, in order to exploit the spatial redundancy, the pixels, surrounding a MB, are considered as reference, given that they are, statistically, the most likely to be very similar to the pixels to encode. In the Inter Modes, the compression is obtained reducing the temporal redundancy, and the reference pixels are the ones belonging to previous and following frames. In order to find the reference pixels in another frame, a search area needs to be defined. The Search Area is the one with the highest probability of being the most similar to the MB to encode, and it is placed in the same position of the current MB, but in a previous (or following) frame.

Figure 2.8 shows an example of a MB in a frame and of the search area in the reference frame, considered for the estimation.

Figure 2.8: Example of the searching area of the previous frame considered for the Inter prediction.

In the following subsections, all the possible modes that can be considered for the Inter prediction, according to the standard AVC/H.264, will be listed and explained.

Inter Mode MB partition

In a video sequence, the temporal redundancy is highly dependent on the motion of the scene. When a MB is in static conditions or with uniform motion, the best reference area of another frame has usually the size of a whole MB with similar motion characteristics. When the motion is fast or, more in general, not uniform, it is more difficult to find a 16×16 pixel area that would properly approximate the MB to encode. In order to maximise the chances of reducing the residual to compress, the Inter prediction modes have been designed in order to adapt the prediction strategy according to the motion characteristics of the scene.

When the area to encode is static or with uniform motion it will probably have a 16×16 pixel size. Otherwise, when the motion of the MB to encode is not uniform it is probable that there are fragmented and not contiguous areas of the reference frame which, all together, approximate the one to compress better than a fully connected 16×16 block. For this reason, in AVC/H.264, it is possible to segment the MBs into several sub-blocks of different sizes and apply the search performed by the Intra MD independently for each of them.

In order to increase the chances of dividing a MB into partitions that suit properly the motion characteristics, several rules, called Inter Modes, have been defined. Figure 2.9 shows how a 16×16 pixel MB can be divided according to the possible Inter Modes in order to follow different kinds of motion. The 16×16 mode is obviously the optimal one when the area is static or the motion is completely uniform for the whole MB. The other modes, instead, try to follow certain motion patterns to increase the possibility of minimizing the difference between the area to encode and the reference area. It is possible to observe that the 8×16 mode is designed to divide the MB vertically into two equal parts to improve the prediction when the left side of the MB has a motion different than the right one. The same happens, horizontally, in the case of the 16×8 mode, where the MB division is applied horizontally to better approximate a MB when the motion of the top part is different than the bottom part. Finally, in the 8×8 mode, the MB is divided into 4 equal parts: the 4 parts are called 8×8 sub-blocks and are encoded independently from each other; every 8×8 can be encoded considering additional Inter Modes.

Figure 2.9: Partitions of a 16×16 MB: 16×16 , 8×16 , 16×8 , 8×8 .

When the 8×8 mode is the chosen one, the 8×8 sub-blocks are encoded independently from each other and it is possible to further divide each of the them to provide a wider amount of possibilities and, as consequence, better approximate the area to encode. The 8×8 sub-blocks can, indeed, be divided following four additional modes. Figure 2.10 shows the possible four further modes, as for the previous ones, when the motion of an 8×8 sub-block is uniform, there is not additional division applied, and, when the sub-block motion is not uniform, this is divided vertically, horizontally, or into four additional 4×4 sub-blocks.

Thanks to this fractal division of the MB, it is possible to separate from each other the areas with different motion. In the next images, a practical example of how this division is performed on a real frame is shown. Figure 2.11 shows two frames of a video sequence and the residual between the two of them: the residual, is the difference between the pixels values of the two frames and, in this case, is used to visually represent the motion of the scene. Figure 2.12 shows the same residual plus the modes that the Inter MD

Figure 2.10: Partitions of an 8×8 sub-block: 8×8 , 4×8 , 8×4 , 4×4 .

stage decided to apply to better separate areas with different motion. It is possible to notice that uniform grey areas in the residual image (which means static or with uniform motion) are all encoded with the 16×16 mode. More dynamic parts, that are represented with darker or brighter grey values, instead, require other modes depending on the nonhomogeneity of the area. A vertical edge, like in the sleeve of the person at the left side of the image, requires in fact the 8×16 . Highly dynamic areas, like in the faces of the two people represented, require smaller segmentation until 4×4 .

2.3. Motion Estimation

The previous section stated why the selection of an Inter Mode is highly dependent on the motion of the scene: the size of the area to predict is related to the motion homogeneity, or nonhomogeneity, of the MB to encode. This means that, when a MB is static or with uniform motion, a larger size mode is more likely to be chosen. When, instead, a MB is affected by strong motion, a smaller size is probably selected.

However, independently from the size of the block to be encoded, each MB, or sub-blocks, is predicted from a same size area belonging to the Reference Frame.

The AVC/H.264 encoder is designed to predict each partition by searching, within the reference frame, the area that better represents the MB, or the sub-block, to encode. Through this searching, the system guarantees the use of the best trade off between the residual and the distortion of the reconstructed video signal at the decoder side. This process is called Motion Estimation (ME).

In order to provide a practical example of how the ME stage is performed, a graphical representation of the search of the best reference for a certain pixel area is shown in Figure 2.13. The Current Frame is represented on the right side of the figure, where the area bordered in black is the MB that is being processed. On the left side, instead, the Reference Frame is represented, where a candidate reference MB has to be found. This reference MB is subtracted to the current one and a residual MB is obtained. The more similar the match, the smaller the residual and the more efficient the compression. In the figure, the two black bordered areas of the Reference Frame contain the part that better matches the MB of the Current Frame and one of them will probably be selected as best match.

Frame 1

Frame 2

Residual

Figure 2.11: Residual Image of the difference between Frame 1 and Frame 2.

Figure 2.12: Residual image of the difference between two frames with the corresponding Inter modes selected for the compression.

Figure 2.13: Example of search of the best matching on the reference frames for one MB of the current frame to be encoded.

2.3.1. Motion Estimation in the Inter Prediction

The ME process is performed iteratively for each possible Inter Mode. This means that, to select the best Mode, the AVC/H.264 encoder needs to perform independently the search for all the sub-blocks the MB is divided into. In addition, depending on the Mode to be tested, the size and the shape of the block will change and, as consequence, also the size and the shape of the area to match will change in the same way.

For example, for the 16×16 Inter Mode, another 16×16 area of the reference frame will be selected, testing all the possible 16×16 areas in a defined search window. The area that gives the minimum residual will be chosen as reference block. When, instead, the system needs to check the compression performance of a smaller partition, smaller matching candidate areas will be tested.

In order to give an example of how the search of a matching block changes depending on the selected Mode, we can consider the same scene provided in Figure 2.11 and in Figure 2.12. Observing the blocks, it is possible to have a practical understanding of which would be the size of the candidate to be found in the reference frame in a real situation.

Figure 2.14 shows an area of the 2nd frame that has been highlighted providing a good representation of nearly all the possible Inter Modes: the top-left MB is being encoded using the 16×16 mode, the top-right and bottom-left are being encoded, respectively, using the 16×8 and 8×16 modes and, finally, the bottom-right is encoded using the 8×8 mode.

Figure 2.14: Four MBs area of a frame to encode with 4 different Inter prediction modes.

In this specific situation, the search of the top-left MB is done looking for a candidate to encode the whole 16×16 MB. For the others, instead, the search is performed independently for the sub-partition to be encoded. This means that for the bottom-

left MB, where the Inter Mode considered is 8×16 , the searching will be performed twice, once for the left 8×16 block and then for the same size block placed on his right. Figure 2.15 shows how the correspondent 8×16 blocks chosen as best matching on the reference frame are found in two positions that are not adjacent because of the motion of the scene.

Figure 2.15: Example of the best matching area found for two 8×16 sub-blocks.

2.3.2. Motion Vectors

Every time that a matching block of the reference frame is chosen, the encoder needs to store the information about the position of the found area. In AVC/H.264, this information corresponds to the difference, in terms of pixel position (p_x, p_y) , between the position of the MB (or sub-block) to encode, in the current frame, and the matched area, in the reference frame. The pixel position difference, is called Motion Vector (MV).

A clear and simple example of a MV is given in Figure 2.16. The black area is a 4×4 pixel sub-block of one MB in the current frame and, besides on the right, the grey region is the same size area of the reference frame used for the prediction. It is possible to see that the shift between the two areas is represented by the arrow pointing to one pixel shift to the right side (+1 pixel toward the x direction) and, as well, to one pixel shift to the upper side (-1 pixel toward the y direction) creating a MV with a value of $(1, -1)$.

The example just provided is a very basic situation. It may happen, however, that given the motion of the scene, the best matching area does not have an integer position in terms of pixels. This is the reason why the ME is also performed at a half, or even a quarter, pixel position resolution for luminance MVs.

When the half pixel resolution is considered, the ME is performed on the values generated by interpolation in the middle positions between integer pixels. Figure 2.17 shows an example of half and quarter resolution pixel matching. The reference block is horizontally shifted by -0.5 pixels with respect to the current one, indicating a half pixel

Figure 2.16: Example of a MV with magnitude (1, -1).

shift, and, at the same time, it is vertically shifted by 0.75 pixels, needing a quarter pixel resolution correspondence.

Every half pixel resolution value is calculated by a six taps FIR filter. For example in Figure 2.18, pixel "b" is calculated considering a weighted interpolation from the pixels E, F, G, H, I, J with weights (1/32, -5/32, 5/8, 5/8, -5/32, 1/32) respectively:

$$b = \text{round} \left[\frac{E - 5F + 20G + 20H - 5I + J}{32} \right] \quad (2.1)$$

In the same way, all the other half-pel position pixels (aa, bb, cc, dd, etc.) are evaluated considering the values of the same row or column so weighted.

When the pixel to be evaluated is placed between four integer pixels, like the pixel "j", it can be evaluated in the same way considering the vertical, or horizontal, half-pel position pixels already evaluated. Using the vertical or horizontal interpolation does not produce any difference.

Figure 2.17: Example of a MV with magnitude (0.75, -0.5).

Once all the half-pixel position pixels are obtained, also the quarter-pixel position values can be calculated. In this case, the evaluation is a simple bilinear interpolation. Figure 2.19 shows how the bilinear interpolation generates the quarter precision value of some pixel interpolated vertically, horizontally and diagonally. For example, pixel "a" is evaluated as in the following equation:

$$a = \text{round} \left[\frac{(G + b)}{2} \right] \quad (2.2)$$

Figure 2.18: Interpolation of luminance half-pel positions.

Figure 2.19: Interpolation of luminance quarter-pel positions.

2.3.3. Motion Vectors prediction

When a MV is encoded, depending on the motion of the scene, the vector representation may need a variable amount of bits: vectors with small magnitude will need be represented by a reduced amount of information, while big magnitudes will need larger amounts. In order to reduce the data to compress, it is possible to try to estimate with a good precision the MVs and reproduce the same process at the decoder side. In this way, it is only needed to encode the difference between the predicted and the actual MVs.

The information used to predict the MV is obtained taking into account the regions of the image that are close to the current one, in order to increase the chances of having a similar motion, and that were previously processed. Figure 2.20 shows an example of the information used to predict a MV. The MB "E" is the one to encode; MBs "A", "B", and "C" are already encoded using the mode Inter 16×16 , so they all have one MV for each MB. The MV of the MB "E" will be predicted as the median value between the set of MVs: MV_A, MV_B, MV_C .

Figure 2.20: A, B and C MBs MVs used for the prediction of the MV of the MB E.

In the example given in Figure 2.20, the surrounding MBs are all encoded using the Inter 16×16 mode, so it is easily deducible which MVs will be considered. However, in many cases, the surrounding MBs are encoded using different Inter Modes as, for example, in Figure 2.21. In this case, the three MVs considered will be belonging to:

- Upper right partition of the "A" MB
- Lower left partition of the "B" MB
- Lower left partition of the "C" MB

These MVs will be always chosen following the above defined three rules and will not depend on the conditions of the MB.

2.3.4. Inter Skip Mode

In a video sequence, it can easily happen that a frame contains extended static or uniform areas like static backgrounds, or objects with uniform motion. In this case, the information for the encoding of each MB may be redundant. To avoid the storing (or

Figure 2.21: A, B and C sub-blocks MVs used for the prediction of the MV of the MB E.

transmission) of useless, or redundant, information, the Skip mode has been included in the AVC/H.264 standard.

When the Skip Mode is considered, the only data encoded is the identifier of the Mode, that consumes 1 bit, not considering the ME data (MVs or residual). At the decoder side, the MV is created calculating the media as in the previous section, and no MV residual is added.

2.4. Bidirectional prediction

The Bidirectional prediction is not considered in the research work presented in this thesis. However, for completeness, it deserves a short section to describe its characteristics.

When the Bidirectional prediction is considered, a MB can be predicted from one or two reference frames, before or after the current one, in temporal order Figure 2.22. This technique allows more possible references, increasing the chances of finding a more accurate prediction and, then, reducing the residual and improving the compression efficiency.

In terms of ME and MV evaluation, the principles of the Bidirectional prediction are similar to the Inter prediction.

2.5. Conclusions

The tutorial in this chapter has described the basis of the MD and ME stages in order to give an overall overview of the technologies considered in the next chapters and to facilitate the understanding of the research work presented.

Chapters 4 and 5 will introduce the development of low complexity strategies focusing on the reduction of the processing time of the MD and ME stages of an AVC/H.264-based compression of sequences in an MVD environment.

Figure 2.22: Three different examples of Bidirectional prediction: (a) past/future, (b) past, (c) future.

3 The Human Visual System and the Perceptual Video Coding - A brief Introduction

Chapter abstract

A different aspect of the MVD video compression is the preservation of the perceptual quality of the synthetic views generated from the encoded depth information. This chapter provides an overview of the human eye characteristics and of the modeling of the human visual system and, finally, presents the fundamentals of the state of the art of the methodologies used to design video encoder based on the visibility thresholds, known as just noticeable differences. It can be considered as an introduction to the research topic presented in Chapter 6.

3.1. Introduction

The Human Visual System (HVS) has been subject of study for many different scientific fields (anatomy, physiology, etc.). Depending on the field, the approaches attempting to develop a knowledge about the HVS have been different. By the video compression point of view, the most interesting observations that can be carried out are about the physical interaction between the light and the matter forming the human eye, the electrical signals transmitted into the nerves and, especially, the psychology of the perception [20].

A knowledge, in depth, of how the human eye perceives an image, can be exploited to generate an accurate modeling of the HVS regarding all the stages affecting the journey of the light, from the cornea to the brain (color processing, sensitivity, masking, etc.).

In the end, encoding algorithms can be optimized taking advantage of the HVS characteristics. Thanks to this modeling, in fact, the compression can be performed, not only trying to preserve the quality of the image, but also trying to preserve the HVS interpretation of it.

In this chapter, it is possible to find a brief description of the interaction between light and the matter forming the human eye in Section 3.2. The HVS description is followed by the fundamentals of its modelling in Section 3.3 and finally, by a short explanation of the Just Noticeable Differences concept, in Section 3.4, that is the fundament needed for the understanding of the research work described in Chapter 6.

3.2. The Human Visual System

In order to have a clear understanding of how the perceptual video coding works and gives effective results, we need to provide a brief overview of the Human Visual System and of how the light interacts with the human eye and the human brain. Figure 3.1 shows a simplified representation of the cross section of a human eye.

Figure 3.1: Human eye cross section.

The first interaction between the light and the HVS happens in the cornea, corresponding to the external surface of the eye. The part of the light that is refracted by the cornea enters the eye through the iris and it is projected to the retina, through the lens.

The iris controls the aperture of the lens, which diameter changes depending on the prevailing light level. Then, the light goes through the lens of the eye, that is able to change its own shape in order to focus the image. Finally, on the back of the eye, there is the retina. The retina, with its light sensitive photo-receptors, translates the electromagnetic energy of the light into the electric signals used by the nervous system.

In the retina there are the photo-receptors, divided into two classes: rods and cones. Rods are sensitive to the luminance level and do not contribute to color vision. Colors are indeed absorbed by the cones that, in addition, are responsible of the ability of seeing fine spatial details. Cones are divided into three types sensitive to different wavelengths. Thanks to the "trichromacy", that is the ability to match any color with a mixture of three suitably chosen primaries, it is possible to describe the color stimulus in terms of three numbers indicating the absorption of light of each of the three types of cones photo-receptors. The more sensitive cones are concentrated in a central region (the fovea) which means that high-resolution color vision is only achieved over a small area at the center of the field of view.

The photo-receptors transform the light properties into electrical signals for the visual cortex of the brain, this is the part of the cerebral cortex responsible for processing visual information.

3.3. HVS modeling fundamentals

Once how the eye processes the light and converts it into an image for the brain is understood from a physiological point of view, it is possible to design a modelling of the HVS and handle it as a reproducible and implementable process.

Figure 3.2: Stages of the HVS modeling.

The HVS is normally modelled as a sequential process that involves 5 different stages:

- Color Processing
- Multi-Channel Decomposition
- Contrast and Adaptation
- Contrast Sensitivity
- Masking

The 5 stages, depicted in Figure 3.2, are briefly introduced below.

Color Processing: the color processing is the first stage of the HVS. At this stage, the image is converted into a perceptual color space based on opponent colors. Afterwards, the information is divided into three channels, one achromatic and two chromatic channels.

Multi-Channel Decomposition: in an HVS modeling, the multichannel decomposition is normally created by a multi-resolution filtering or a wavelet decomposition. The reason why this kind of methods are used is that the HVS is based on multiple channels sensitive to different spatial frequencies and orientations.

Contrast and Adaptation: the HVS is also sensitive to local variations and to background, or foreground, illumination differences; contrast levels are used to measure such local variations. The perceived contrast, in addition, is influenced by the ability of the human eye to adapt to a specific illumination level.

Contrast Sensitivity: in order to model the human contrast sensitivity, the Contrast Sensitivity Functions (CSF) have been implemented. The CSF are defined for each channel of the color space (chromatic and achromatic) and also for the temporal sensitivity.

Masking: the word Masking is used to represent the situation when a stimulus, that is normally perceived, cannot be perceived due to the presence of another stimulus. Masking can happen in several situations and it is normally defined in a different way depending on the conditions that generates it, i.e. Contrast Masking, edge Masking, Texture Masking. It also happens an effect opposite to masking, called facilitation. That is defined as the effect that occurs when a stimulus that is not visible by itself can be detected due to the presence of another.

3.4. Just Noticeable Difference (JND)

The studies about the HVS, in addition to describing the biological process and the modeling of how the human eye works and how brings the information to the brain, have also highlighted the limitations. The human eye, in fact, is not able to perceive all the changes in an image, due to many physiological and psychological mechanisms. Knowing these mechanisms is a great advantage in the design of image processing techniques, because they can be exploited in order to properly allocate the limited resources, and obtain perceptually improved performance. In particular, the knowledge of the HVS sensitivity, permits a modelling of its limitations that can be translated into visibility thresholds. Incorporating these visibility thresholds within any image processing algorithm is very important to optimize them according to the human eye. The Just Noticeable Differences (JND) are one of the most common visibility thresholds considered in video coding. As the JND definition [20] says, "A just noticeable difference refers to the threshold below which any change cannot be detected by the HVS".

This section shows an overview of the studies, provided by Wu and Rao [20], of how the JND can be modeled in order to be used in an image or video compression algorithm. The JND can be formulated in several ways as, for example, function of the Discrete Cosine Transform (DCT) that, as it is widely known, is used to reduce the correlation of the image data in many of the video compression standards. It can be as well formulated in the pixel domain, as the perceptual coding work presented in this thesis (Chapter 6).

3.4.1. JND with pixels

The pixel domain JND has been created to apply the JND concept on applications where the use of DCT based effects is not possible. Examples of exclusive pixel domain manipulations are motion estimation or specific pixel values quality evaluations.

The pixel domain JND is normally evaluated considering its spatial and temporal effects. For the spatial pixel based JND, situations as background variations or neighbour pixel value changes will affect the HVS in perceiving an image or a video. For the temporal one, the limitations are due to the inter-frame variations. So, in both cases, the pixel domain JND are affected by activities in the surrounding areas.

Spatial JND (SJND)

In the Spatial Just Noticeable Differences (SJND), background luminance and neighbour pixels variations are the effects to take into account. This Spatial JND of the pixel (i, j) can be modelled as:

$$P(i, j) = T^L(i, j) + T^t(i, j) - C^{Lt}(i, j) \cdot \min(T^L(i, j), T^t(i, j)) \quad (3.1)$$

where $T^L(i, j)$ and $T^t(i, j)$ are the visibility thresholds for luminance adaptation and texture masking for the pixel (i, j) , respectively. The value $C^L(i, j)$ is a weighting factor

between $[0, 1]$ used to take into account the effect of overlapping between $T^L(i, j)$ and $T^t(i, j)$.

The luminance adaptation visibility threshold $T^L(i, j)$ can be approximated as:

$$T^L(i, j) = \begin{cases} 17 \left((1 - \sqrt{\frac{L(i, j)}{127}}) \right) + 3 & \text{if } L(i, j) \leq 127 \\ \frac{3}{128}(L(i, j) - 127) + 3 & \text{otherwise} \end{cases} \quad (3.2)$$

where $L(i, j)$ corresponds to the average background luminance at position (i, j) evaluated over a certain area of $N \times N$ pixels. Figure 3.3 shows the luminance adaptation visibility threshold in function of the average background luminance in an 8 bit gray-scale image representation $[0, 255]$.

Figure 3.3: Visibility threshold $T^L(i, j)$ in function of the average background luminance $L(i, j)$.

In turn, the texture masking visibility threshold can be evaluated as:

$$T^t(i, j) = \beta \cdot g(i, j) \cdot e(i, j) \quad (3.3)$$

where β is a control parameter; $g(i, j)$ represents the weighted average of gradients around the pixel in position (i, j) ; $e(i, j)$ is an edge-related weight of the pixel at position (i, j) , and its corresponding matrix e is computed as:

$$e = \hat{b} * h \quad (3.4)$$

where $\hat{b}$ is based on the type of edge of the original image, with possible element values of 0.1, 0.3 and 1.0 for smooth, edge and texture pixels, respectively; h is an $m \times m$ Gaussian low pass filter with standard deviation σ .

Tuning the weighting factor, $C^{Lt}(i, j)$, that takes into account the effects of the overlapping between the two visibility thresholds, ($T^L(i, j)$ and $T^t(i, j)$), it is possible to generate a simplified version of the JND. Setting a value $C^{Lt}(i, j) = 1$, only the major effect between the two contributors (luminance adaptation and texture masking) is considered:

$$P(i, j)_{\text{simplified}} = \max(T^L(i, j), T^t(i, j)) \quad (3.5)$$

Temporal JND

As for the spatial masking, also the temporal variations have effects over the JND values. Usually, bigger inter-frame differences lead to larger temporal masking.

The temporal JND can be modelled as:

$$P_v(i, j) = f(d(i, j)) \cdot P(i, j) \quad (3.6)$$

where $f(d)$ is the empirical function to reflect the increase in masking effect with the increase in inter-frame changes, as shown in Figure 3.4, where $d(i, j)$ represents the average inter-frame luminance difference between the current frame x and the previous frame x^p , evaluated as:

$$d(i, j) = \frac{x[i, j] - x^p[i, j]}{2} + \frac{L_0 - L^p_0}{2} \quad (3.7)$$

where L_0 and L^p_0 are the average luminance of x and x^p , respectively.

Figure 3.4: Temporal masking effect function $f(d)$ in function of the average inter-frame difference $d(i, j)$.

3.5. Conclusions

This chapter has given an overview of how the HVS limitations can modeled by the JND and how they can be exploited in order to optimize the video compression process. The JND, evaluated on the pixels domain, is at the basis of the work developed and presented in Chapter 6 where, indeed, the same concepts are moved and, in some case, translated to the MVD environment for the perceptual optimisation of the encoding process of a content that usually is not provided to the observer: the depth sequences.

4 Fast Mode Decision for Multiview Video Coding based on scene geometry

Chapter abstract

The Mode Decision is one of the heaviest stages in video compression. In an MVD environment, given the high amount of video and depth sequences, the complexity of the MD affects the latency of the encoding process in a much heavier way than in traditional video coding. The work presented in this chapter introduces an algorithm able to reduce the complexity of the MD stage of an AVC/H.264 encoder working in an MVD environment. Exploiting the information available in a 3D video scenario, represented by the depth, that provides a knowledge of the distance of the objects to the camera, and exploiting the strategies used for the compression of other views (previously encoded), it is possible to considerably reduce the computational load of the MD stage. The idea behind this study is that areas with different depth should have different motion characteristics. Dividing every Macro Block (MB) into sub-blocks according to the depth homogeneity, the areas with different motions should be properly separated and, at the same time, a considerable amount of operations can be avoided, given that the MD stage does not need to analyze all the available modes. The study has been divided into two approaches. In the first one, the depth characteristics, as local homogeneities and differences, have been exploited to define the best Mode to encode the MB. Using the depth, it is possible to avoid checking all the possible Modes defined in AVC/H.264, however, in certain situations, the depth does not give an accurate estimation of the motion characteristics, for this reason a crossed double checking is needed. Such checking is performed exploiting the encoding information of a neighbouring view, called Reference View, previously compressed. In the second approach, the disparity between views has been taken into account. In the first approach, the Reference View motion has been considered without any correction of the disparity. In this one, the displacement between the Reference View and the sequence to encode is included. These two proposed approaches reduce the computational load of the MD stage for the encoding of a texture sequence in a multiview scenario saving up to 40% of the encoding time compared to two of the MD algorithms implemented in the AVC/H.264 reference software, Low Complexity MD and High Complexity MD. In terms of rate distortion, the proposed strategies, in most of the situations, reaches the best performance of the two algorithms used for the comparison.

4.1. Introduction

As explained in Chapter 1, the scene can be recorded by two or more cameras in a 3D video environment. In addition, for every camera, the traditional video (called texture

sequence) is enriched by a corresponding depth sequence, providing the information about the geometry of the scene. The large amount of information to process implies an increase of the number of operations needed for the compression.

To solve this issue, this work focuses on the complexity reduction of the Mode Decision (MD) stage of the texture sequences compression in an MVD environment. The goal of the MD is to find an efficient, block-based, approximation to the video motion field, considering the rate-distortion cost of every possible strategy. What ideally happens is that different block sizes are chosen so that a single motion vector provides an optimal motion compensated prediction for all pixels in the block. However, when all the possible modes and MVs have to be tested, the number of operations needed to choose the best selection to encode the sequence is obviously huge and, considering that in a multi-view environment the number of sequences is considerably high, the total computational burden would be hard to process in a reduced time.

The work presented in this chapter proposes to exploit the information included in a 3D environment, such as multiple views and depth sequences, not only for the 3D reconstruction, but also for the reduction of the amount of operations needed for the compression. The presence of the depth, indeed, provides an additional representation of the objects of the scene: the distance from the camera, that can be also exploited to divide the frames according to the motion.

Considering that different objects have often different motion characteristics and, at the same time, are also represented with different depth levels, the proposed algorithms apply a block division based on the depth differences using i) big block sizes for areas that, in terms of depth, are homogeneous and using ii) smaller block sizes for areas less homogenous.

On the other hand, there are some situations that make not always reliable a motion description given by the depth differences. For example, there could be areas of the same object which are represented with the same depth but have different motion characteristics. There could also be areas with depth differences that are static or with homogenous motion. To overcome this issue another additional information, as the motion information already evaluated for other views, is exploited. Using a sequence recorded by a camera placed in a close position, and previously encoded, a description of the motion of the scene is provided for free and can be used to estimate when the depth based MD is reliable and when it is not.

This work has been developed in two stages. First, the depth homogeneity has been used for the mode selection, considering also the MD previously evaluated for a contiguous view. After, the method has been improved considering also the disparity between views, reaching better results.

The rest of the chapter is organized as follows: Section 4.2 shows a brief introduction of several low complexity algorithms proposed for the encoding of 2D and 3D video sequences. Section 4.3 describes in detail the first part of this study. In section 4.4 the second part of this research is presented, where the disparity between views is considered. Section 4.5 shows a numerical analysis of the complexity reduction obtained by the two algorithms and, finally Section 4.6 provides the results obtained by the two MD methods developed.

4.2. Related work

As explained previously, in a multiview environment, the processing time could be a relevant issue. Due to the presence of several captures and to the presence of the depth sequences (both needed for the 3D reconstruction), the overall computational load can drastically increase, requesting a huge processing time.

In this scenario, fast MD algorithms can provide an important reduction of the complexity at the encoder side, bringing, as a consequence, also a reduction of the time needed for the video compression.

Also for traditional video coding, due to the complexity of the MD stage, the research focused on the computational reduction mitigating the quality loss. Many MD algorithms, referenced in the literature, are based on alternative Rate-Distortion (RD) computation methods [21], early stop criteria [22] or on particular prediction techniques [23]. After the incoming of the 3DTV, a branch of the research have specifically focused on how to reduce the complexity of the multi-view video sequences encoding.

One of the easiest way to reduce the complexity is exploiting the additional information available in the multi-view scenario. In the work of Ding *et al.* [24], for example, the disparity between two views is used. The Disparity Vectors (DVs) between two views are used to find a correspondence between the MB of the view to encode and an area of a neighbouring view used as a reference. Once this area is found, the previously evaluated MD, of the reference view, is overlapped on the MB to encode. Using this method, the MD is provided without the need of any operation.

A similar approach is used in the work of Han *et al.* [25]. In this case, the considered DVs are not the local disparity vectors, but the global ones, which are defined as the whole shift between two different views. Han proposes to create a subset of modes, called background modes corresponding to the biggest partitions. Using the DVs, the method is able to separate foreground MBs from background MBs and, if a MB is considered in a background area, the MD is evaluated only among a reduced subset of modes.

The disparity is exploited also in the study of Peng *et al.* [26]. In this case the mode considered for the MB, that overlaps the corresponding area of the nearest view, is included between the candidates of the Rate Distortion Optimization (RDO), in addition to Intra and Skip modes. This method outcomes the performance of the previous ones in terms of RD quality. However, this happens with a lower computational reduction.

The work of Zhu *et al.* [27], instead, follows a different approach. Given two neighbouring views and given the correlation between them, the RD cost evaluated for one view should be very similar to the other. This information can be exploited to provide an early decision about the Skip mode. In fact, when the RD cost of the Skip mode evaluated for a MB is similar to the cost evaluated for the same MB in the neighbour view, the evaluation of all the other modes can be skipped.

Also in the work of Shen *et al.* [28] the correlation between neighbouring views is exploited. In this case the early decision is done considering the complexity of the mode selected for the corresponding MB of the neighboring view. Shen divides the Inter modes in three degrees of complexity depending on the number of sub-blocks used

to divide the MB: no complexity for Skip and 16×16 , low complexity for 16×8 and 8×16 and high complexity for 8×8 and sub-divisions. When the MB of the neighboring view is compressed using a high complexity mode, all the possible modes are tested for the MB to encode. When, instead, it is encoded through a low complexity mode, the 8×8 mode and its sub-divisions are not considered, avoiding the evaluation of a considerable amount of operations. The computational load is further reduced when the neighboring view MB is encoded through a mode among the no complexity ones: in this case only Skip and 16×16 are considered.

Lee *et al.* [29], instead, considered different strategies depending on the presence of motion in the foreground and in the background. Knowing that foreground objects, with quick motion, are often predicted by a disparity compensated prediction mode, dividing foreground from background, applying a segmentation, it is possible to perform an object based early stop mode selection. The number of candidates is then reduced depending if a MB belongs to the background or not.

All these methods obtained a considerable reduction of the computational complexity needed for the MD evaluation. However they have paid this reduction in terms of quality. The work presented in this chapter is based on a previous study of Cheung *et al.* [30] where the depth information provided in a video game environment can be used to reduce the processing time of the MD stage for the compression of synthetic content in networked online games. The proposed method is able to considerably reduce the computational burden of the MD stage without significant losses in terms of quality. The algorithm considers the information given by the depth sequences, taking into account that there could be a correlation between the motion and the depth of a sequence.

The complexity of the MD can be reduced if the boundaries between regions with homogeneous motion are known. As proposed by Cheung *et al.* [30] for computer generated video coding, it is possible to notice that uniform motion regions are less likely to occur in areas where the depth field is non-uniform. The depth information, then, can be used to estimate the structure of the motion field. The main differences of the proposed algorithm with respect to Cheung *et al.* [30] are that (i) the problem of using depth information for the MD is addressed to the context of multi-view video sequences, and (ii) other additional information, provided in a multiview scenario is used: the disparity between views.

4.3. Depth based Fast Mode Decision (DFMD)

In the AVC/H.264 based video coding process, images are divided into MBs: 16×16 pixel regions considered as basic and independent units for the motion compensated prediction. Every MB can be divided into sub-blocks, chosen among a range of sizes from 16×16 down to 4×4 , in order to improve the compression efficiency. If the motion of a MB is homogenous, large partitions are more likely to be used, and, vice versa, non-homogenous motion tends to lead to smaller block-sizes.

As explained in section 2.2, the process used to test all the possible strategies (called Modes) and to select the best one, in terms of RD cost, is the MD.

The proposed study aims, initially, to exploit the correlation between the motion and the depth information, in order to design a method able to reduce the complexity of the MD stage. In fact, if the selection of the best mode is evaluated considering the depth homogeneity of the scene, it is possible to avoid to check all the possible strategies.

The method derived from this study is called Depth based Fast Mode Decision (DFMD), and it is able to reduce the complexity of the MD process by exploiting two kinds of information: the depth map of the view to encode, called Current View (CV), and the motion-based MD of a view recorded by a camera placed in a close position, called Reference View (RV), which has been previously encoded.

Figure 4.1: Overall architecture of the DFMD algorithm.

Fig. 4.1 provides an overview of the overall architecture of DFMD. The process starts evaluating the threshold that links the motion domain to the depth domain. Given that the correlation between depth and motion may change depending on the several situations, a threshold is needed for each MB. Such threshold defines, with a certain accuracy, which depth difference corresponds to a motion difference and, on the other hand, which depth difference, instead, corresponds to areas with uniform motion. The threshold evaluation stage is described in detail in Section 4.3.1. When the threshold is greater than zero, the method jumps to the following stage: "Depth map based MD". At this stage, the image to compress is segmented in function of the depth homogeneity. We assume that regions that have homogenous motion are also more likely to have homogenous depth. For example, an object moving in front of a static background would have a lower depth value than the background region. According to this concept, boundaries between regions which are depth-wise homogeneous should correspond to boundaries in the motion field. The proposed method exploits this correspondence between depth and motion region boundaries. Section 4.3.2 describes this step more in detail.

When, instead, the evaluated threshold is zero, there is not a depth difference that can be linked to a motion difference. In this case, the RV is used to provide a further estimation of the motion of the considered area. Section 4.3.3 describes how these situations are managed.

4.3.1. Threshold computation

The threshold computation is the first stage of the DFMD process and it is needed to define an adaptive link between the depth domain and the motion domain and changes for each MB. An MD, based on a fixed depth homogeneity, in fact, may lead to inaccurate results: when depth differences do not have a correspondence in the motion field, complex modes could be applied to static MBs and, on the contrary, large partitions could be applied on MBs with strong motion. In order to improve the efficiency of the MD, the algorithm needs to manage the mentioned specific situations in different ways. The proposed threshold evaluation is based on the temporal analysis of the most reliable information about the motion of the scene: the RV, which has been already encoded and is available without any further computation.

The threshold represents an estimation of the minimum depth difference that will trigger a division of the block into sub-blocks in the following stage: "CV Depth Map based MD" described in Section 4.3.2.

For this stage we consider:

- The MB to encode, belonging to the CV texture: Current MB.
- The MB of the CV depth map placed in the same position of the Current MB: Current Depth MB.
- The MB of the RV texture, placed in the same position of the Current MB: Reference MB.

Figure 4.2: Flow diagram describing all the stages of the threshold evaluation.

The partition evaluated by the MD stage of the Reference MB is applied on the Current Depth MB, dividing it into sub-blocks. Fig. 4.3 shows how the RV mode is used to divide the depth MB for the CV.

Figure 4.3: Example of the overlapping between the RV selected mode and the depth MB for the CV.

Thanks to this partition, the algorithm is able to detect which depth differences correspond to motion variations. In order to provide a numerical evaluation of this depth

differences, the most representative depth value must be selected for each sub-block. If an object is moving, the points with higher probability to have higher motion are the closest to the camera and, as the maximum depth value, for each sub-block, represents that closest point, then it is used as the most representative value for each area. As every sub-block should have different motions (information provided by the RV MD), every maximum depth value should represent a different motion area. In order to perform a partition, the minimum depth difference is needed to compare the depth sub-blocks and to divide the MB. In this way, all the differences between the maximum depth values, previously evaluated, are calculated and the minimum one is selected as a threshold:

$$th_k = \min |d_i - d_j| \quad (4.1)$$

$$i, j = 1, 2, \dots, \# \text{sub-blocks}$$

where th_k is the threshold for the k -th MB and d_i and d_j are the maximum depth values of sub-blocks i, j within the k -th MB (Fig. 4.4). Applying this procedure to the entire image, an adaptive threshold is obtained for each MB.

d_1	d_2	d_3
	d_4	d_5
d_6	d_7	

Figure 4.4: Example of the depth maximum values in a MB with three 8×8 partitions and one 4×4 submode.

4.3.2. Depth Map based Mode Decision

Once the threshold is evaluated, the algorithm has already a knowledge of how the motion of the scene is linked to the depth information. The goal of this second stage is to provide a partition that depends on the depth of the scene exploiting the correlation between depth map and motion. The idea is to divide into blocks every frame of the CV through the analysis of its depth map. Homogeneous areas of the depth map will be selected as areas to compress using large partition sizes, whereas areas with different depth values will be divided into smaller partitions.

The process starts by dividing the depth frame into MBs. The splitting mechanism, described in Figure 4.5, is applied independently for every MB. First, the MB is divided into 4 sub-blocks of 8×8 luminance samples and the average depth value is evaluated for each one of the four blocks. If all the differences between the average values of the whole MB and each one of the sub-blocks (8×8) are smaller than the threshold, the MB is classified as homogeneous. Supposing that homogenous depth areas do not have significant motion, the MB is encoded with the 16×16 mode. Otherwise, the algorithm

Figure 4.5: Flow diagram describing all the stages of the Depth based MB partition for the CV.

continues checking the following partitions: 8×16 and 16×8 . To test them, the 8×8 sub-blocks are compared in pairs. The vertical and horizontal homogeneity is evaluated estimating the difference between pairs of adjacent sub-blocks. If two 8×8 adjacent sub-blocks have similar values, and, at the same time, the other pair of adjacent sub-blocks have also similar values (as the difference is lower than the threshold), then the corresponding intermediate partition is selected. Instead, if the horizontal or vertical pairs of 8×8 sub-blocks are not considered homogenous in depth, then the 8×8 mode is selected.

If the 8×8 mode is the selected one, the finest partitions need to be tested. The algorithm, then, follows the same strategy on every 8×8 block. In other words, the same procedure is applied, keeping the same threshold, on every 8×8 sub-block to test the presence of the 4×8 , 8×4 and 4×4 partitions.

4.3.3. Analysis of the surrounding area

When the depth of the MB of the CV is totally homogenous, it is possible to observe that any partition would provide a threshold equal to zero. In this case, the depth map cannot be trusted to provide the MD because the 16×16 partition would be directly applied without knowing if the motion field is homogenous or not. To solve this problem, when the evaluated threshold is zero, an analysis of the surrounding area of the MB is performed to check if the 16×16 mode is appropriate or not.

Due to the high correlation between the motion fields of CV and RV, this analysis is carried out for the RV MBs.

The considered MBs are those placed around the RV MB placed in the same position of the CV MB to encode (Fig. 4.6). If the 16×16 mode is the strategy selected for

the majority of them, the chance to find a region with homogenous motion (or static) is high: in this case the 16×16 mode turns out to be the best selection. Otherwise, the proposed DFMD is not reliable and a traditional MD algorithm, implemented in the AVC/H.264 Reference Software (RS), is applied.

Figure 4.6: MBs surrounding the current one (shaded) for the 16×16 analysis.

The surrounding area analysis is performed according to the Algorithm 4.3.1. In the pseudo-code it is possible to see that the decision is taken according to the threshold *min_number*. The value used for the variable *min_number* is indicated in the Section 4.6.

Algorithm 4.3.1: SURROUNDING AREA ANALYSIS (*Mode*)

```

count = 0
for i ← 1 to 8
  do { if Mode(i) == 16 × 16
      then count ++
    }
if count >= min_number
  then Mode(0) = 16 × 16

```

4.3.4. Conclusions about the DFMD

This first method shows how it is possible to exploit the depth information in order to perform a segmentation able to reduce the computational load of the MD stage for MVD sequences.

Figure 4.7 provides a final overall description of the work flow followed by the algorithm. It is possible to see that the core of the algorithm is the threshold evaluation. Depending on the value calculated for the threshold, in fact, the algorithm decides if to perform an MD based on the depth information or not. When the threshold is higher than zero, then the depth information is considered reliable and the algorithm moves to the stage represented on the left of the diagram. On the other hand, if the threshold is zero, the RV is analyzed to check the uniformity of the motion of the considered area.

The main limitation of this approach is given by the fact that the RV is used without considering the shift between views. In the following sections, an improved version of this method is presented than takes into account the disparity between RV and CV.

Figure 4.7: Flow diagram of the DFMD Algorithm

4.4. Disparity and Depth based Fast Mode Decision (DDFMD)

As explained in the previous sections, the proposed research work targets a multi-view environment where many kinds of information are available, and it aims to reduce the computational complexity of the MD stage.

Section 4.3 presented how the depth information can be used to provide a good estimation of the motion field. To know which depth differences correspond to motion differences, the neighbouring view, called also RV, is exploited without any further processing. Now, in this section, the accuracy of the motion field given by the RV is improved considering another additional information provided in a multi-view scenario, which is the disparity between views. The method is called Disparity and Depth based Fast Mode Decision (DDFMD).

The DDFMD method, is based on the DFMD and presents some similarities. The part of the algorithm related to the division of every MB into sub-blocks and the part related to the surrounding analysis evaluation are the same. The difference is represented by the use of the Disparity Vectors (DVs) that are used to drive the threshold evaluation. Thanks to the disparity between views, the thresholds are evaluated with a higher precision, and they are more correlated with the motion of the CV. The division into sub-blocks is so more accurate and the performance is improved, as demonstrated in the experimental results.

4.4.1. DDFMD threshold computation

As explained in the DFMD algorithm description (Section 4.3), the splitting process is not static. Due to the not always reliable characteristics of the depth, there is not an absolute certainty that every depth gap corresponds to a motion difference. Therefore, an adaptive mechanism has been created, able to change the sensitivity of the algorithm to the depth differences according to the motion of the scene. The goal of this mechanism is to evaluate, using the RV and the DV, a threshold which should define a limit between the depth differences which correspond to a motion difference and the depth differences which belong to static areas.

In the DFMD, the MD previously evaluated on the RV MB was used to divide the depth MB into different sections with different motion. The threshold was evaluated according to the RV motion and the depth differences (Figure 4.3).

A threshold estimated using the described method, provides good results if there is not disparity between RV and CV or when this is not remarkable. However, when the objects in the two views are displaced, the application of the MD of the RV on the depth map of the CV may provide a bad threshold due to the not perfect overlapping. In Figure 4.8, as an example, it is possible to observe that, applying the RV partition on the CV depth, areas with different motion will be joint in the same sub-block (top left 8×8 block) and areas with uniform motion will be separated (top right 8×8 block).

Figure 4.8: Bad correspondence between RV and CV in case of disparity.

This problem is solved using the DV, that are vectors connecting the corresponding points between the RV and the CV. The CV DV is used to find the corresponding area in the RV for each MB. The displaced RV partition is so overlapped on the depth map exactly on the correct position (Figure 4.9.a). As a consequence, the depth values used for the threshold evaluation are different and give better results because they describe more accurately the motion differences that can be found inside the MB (Figure 4.9.b). The threshold is so evaluated following the rules described for the DFMD (Section 4.4.1).

4.5. Complexity reduction analysis

In the previous sections, a method for the complexity reduction of the MD stage has been presented. The reduction is performed by exploiting the depth information and its correlation with the motion field. However the depth field could not be always reliable

Figure 4.9: Depth map's segmentation using disparity.

for the application of a motion based segmentation, so the number of operations can vary according to the different situations. In this section, the complexity reduction provided by these methods is analyzed.

Traditional AVC/H.264 computational burden

According to AVC/H.264, the mode decision is the process needed to estimate the best division of each MB according to the motion homogeneities. This division is estimated evaluating the distortion with respect to the original MB through several techniques as Mean Squared Error (MSE), Sum of Absolute Differences (SAD) or others, measuring the pixel-wise differences. As a MB is an area of the image covering 16×16 pixels, to compare the computational burdens, it is possible to consider as basic unit the number of distortion evaluations on 16×16 pixels' MB, $D_{16 \times 16}$.

The possible modes for the decision are Intra and Inter, but, in this study only the Inter are considered for the modes reduction, hence the analyzed modes are the eight Inter possible ones: 16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , 4×4 and Skip Mode.

For every mode, in the encoding process the distortion is evaluated comparing the 16×16 pixel MB with an area of the reference frame. Considering the default setting of a full search method, such area is a ± 32 pixel window around the current MB position (65×65 pixels). Therefore the candidates are all the 16×16 pixels blocks placed inside this area, which means that $65 \times 65 = 4225$ $D_{16 \times 16}$ operations are needed. Multiplying this number for the eight possible modes when all the modes are tested we obtain $8 \times 4225 = 33800$ $D_{16 \times 16}$ distortion evaluations are needed.

DFMD and DDFMD computational burden

The DFMD and DDFMD methods are able to detect if the MD can be evaluated measuring the depth differences, but also to recognize if the depth is reliable for the motion field estimation. According to the different situations, the algorithm might be able to drastically reduce the computational burden or to leave it unchanged with respect to the traditional AVC/H.264.

In general, threshold and depth homogeneity are evaluated to detect if the depth can be used to perform the MD. For the depth homogeneity evaluation, the local average depth is calculated. For the evaluation of the average on a 16×16 pixels MB, $(16 \times 16) - 1 = 255$ Av_{sum} sums are needed, plus the division by the number of pixels. For the evaluation of the sub-blocks average values, the division can be done on the partial cumulative sums when the 16×16 is evaluated, so, for example, when the 8×8 block sums are evaluated, the average values can be found just with a division, then the cumulative sum can be added to find the other sub-blocks average. Therefore, beside the 255 Av_{sum} sums, a division Av_{div} for each mode (7 Inter modes) is needed.

For the threshold computation the local maximum must be evaluated. Also in this case, the evaluation is not depending on the number of sub-blocks because the comparisons must be done for each pixel of the MB, so $(16 \times 16) - 1 = 255$ Th_{comp} comparisons are needed.

As explained in Section 4.3.3, when the depth map is not reliable, the RV motion is also used to decide if the MD is performed as in the Traditional AVC/H.264 or not, so the computational burden of each situation can be separated into three different situations:

- Reliable depth
- Non reliable depth and homogenous motion in the RV
- Non reliable depth and non homogenous motion in the RV

Table 4.1 shows the computational burden required in all the situations. In the worst case, just a small burden is added to the number of operations needed by a traditional MD algorithm, but, as shown in the next section (section 4.6), where the computational time is evaluated for each algorithm, this is widely compensated by the other two situations.

	D(D)FMD Algorithm	Traditional AVC/H.264 MD
Reliable depth	$255Av_{sums} + 240Th_{comp}$	$33800D_{16 \times 16}$
Non reliable depth and RV homogenous motion	$255Av_{sums} + 255Th_{comp}$	
Non reliable depth and RV non homogenous motion	$255Av_{sums} + 255Th_{comp} + 33800D_{16 \times 16}$	

Table 4.1: Computational complexity reduction.

4.6. Experimental results

The study described in this chapter presents an analysis about how the depth information can be exploited to perform a low complexity MD in a 3DTV scenario. Two methods, able to reduce considerably the MD computational load are shown: the first one considers the own depth map and the motion field evaluated by a neighbor view; a second one which corrects the limitations of the first one and considers the disparity field to compensate the motion field between views.

As the study aims to reduce the computational load of the MD stage in an AVC/H.264 compression process, the time needed to process the encoding of a set of sequences is measured for the two considered methods, and compared to three alternative fast MD methods.

However, even if the goal is the complexity reduction, the quality of the encoded video must be considered. Therefore, also the RD performance is compared with the alternative methods to check if the quality is preserved.

This section is divided into two sections: first, the experimental setup is described, explaining metrics, encoding setup and sequences considered; second, the results are presented and commented.

Experimental setup

For the performance evaluation, a set of MVD sequences has been selected between the videos evaluated by the Moving Pictures Expert Group (MPEG) [31], with an acceptable evaluation of the quality of the depth sequences. The selected sequences are described in Table 4.2, where the name of each sequence is followed by the resolution, frame rate, and the identification number of the views used as reference (RV) and current (CV) in the test.

Sequence	RV	CV	Resolution	Fr. rate
Beergarden	5	6	1920 × 1080	25
Book Arrival	8	10	1024 × 768	16.67
Cafe	3	4	1920 × 1080	30
Akko&Kayo	27	28	640 × 480	25
Balloons	1	3	1024 × 768	30
Mobile	4	5	720 × 540	25

Table 4.2: MVD sequences used for the experiments.

As explained in the previous sections, the DFMD/DDFM algorithms exploit some information available in a 3DTV environment such as depth, disparity and neighboring views. Depth and disparity are considered already evaluated for other purposes and available for free. The neighbouring view (RV) is considered previously encoded and its MD evaluated through a traditional AVC/H.264 encoder.

The encoder common settings used for the compression of both RV and CV are first described, and then the differences are highlighted. RV and CV views are both encoded using the AVC/H.264 Reference Software (RS) version 18.0 [32]. In all the cases,

4 seconds of video have been used, so the number of frames depends on the frame rate. The type of Group of Pictures (GOP) used is the IPPP... adding an I frame every second of video. B modes might be interesting to observe, but the kind of temporal compression provided is similar to the P frames, so, to simplify the algorithm implementation, the B modes are not considered. A list of the common encoder settings is described as follows:

- JM Reference Software version 18.0 [32]
- AVC/H.264 Main Profile
- One Group of Pictures (GOP) for every second
- GOP structure: IPPP...
- Enabled Modes: Intra and Inter

Using these common settings, the tests have been performed considering four RD points, calculated tuning the AVC/H.264 Quantization Parameter (QP) within the values 28, 32, 38, 42.

In the encoding settings of RV and CV there are few differences related to the MD and ME stages. In the RV, a traditional settings has been used, considering a Full Search MD enabling all the Inter and Intra Modes. However, given the absence of a MB partition, for the MD information interchange with the CV, Intra modes would be useless, so even when the RD optimization stage decides for an Intra mode, only the best between all the checked Inter modes is provided to the CV. In the ME stage, of the RV, a Full Search of MVs is performed on a ± 32 pixel window around de current MB position on the previous frame considered as reference. The Full Search ME is performed for all the modes.

For the CV, the encoder configuration is the same, with the only difference that the MD is performed by the DFMD or DDFMD algorithms for the Inter MD, except for the Skip mode, which is performed in all cases. Intra modes are also enabled due to the high optimized compression in areas with homogenous textures and the low computational load needed for the Intra MD. Therefore, the RD optimization is performed between the Skip mode, the Intra modes and between the mode provided as output by the DFMD (or DDFMD) algorithm. In addition to the aforementioned settings, the CV needs a value for the minimum number of MB for the surrounding area analysis (*min_{number}* in the Algorithm 4.3.1. In the experiments shown in this section the value *min_{number}* as been set equal to 4 so that the computational load is not excessively increased.

To compare the results provided by this study to other methods, the compression has been performed also by a set of MD algorithms. Two of the algorithms used for the comparison are implemented in the reference software and the third one is a simplified, fast and computationally-free version of the MD. They are listed as follows:

- High Complexity MD
- Low Complexity MD
- Re-use of the RV Modes for a computational-free MD

The High Complexity MD algorithm, performs its mode decision checking all the required modes and evaluating the RD cost for each possible MV of the search window for each mode.

The Low Complexity MD algorithm reduces the computational load by avoiding the evaluation of the RD cost for the MVs, and substituting it with a simple Sum of Absolute Differences (SAD) evaluation. In this way, the encoding time is significantly reduced, but the compression quality is not optimized on the RD cost.

Given the correlation between two neighbor views in an MVD environment, the MD evaluated for the RV could represent a good approximation of the one needed for the CV encoding and, in addition, is computational-free. In this way, it is used as a third reference algorithm in the comparison.

For all the comparisons, the RD performance is evaluated using the widely known Peak Signal to Noise Ratio (PSNR) and, as the study aims to reduce the computational complexity of the MD of the MVD sequences encoding, also the encoding time, needed for the compression, is evaluated.

Performance evaluation

The twofold goal of this study is, first of all, the computational complexity reduction of the MD stage in the encoding process of MVD sequences when information as depth and disparity are available and, second, the minimum possible RD performance loss.

In order to evaluate the effectiveness of the DFMD and DDFMD algorithms, the RD performance is measured evaluating the PSNR for the two developed methods and for the three algorithms used for the comparison. In order to evaluate the computational reduction, the encoding time needed for the compression has been evaluated for all the considered cases.

RD performance evaluation

Figures 4.10, 4.11, 4.12, 4.13, 4.14, 4.15 show the comparison of the PSNR performance for the developed DFMD and DDFMD methods, the traditional AVC/H.264 High and Low Complexity methods, and the copy of the RV MD for the six sequences under study. The results show how the Low Complexity MD used in the RS, gives almost always the worse results in terms of RD performance. Only in the case of the sequence Cafe (Figure 4.15), at high bit rates, the Low Complexity MD method is able to improve the performance given by the direct copy of the RV MD.

The MD performed for the RV, according to the results, reveals to be a very good estimation for the CV motion field variations. Just copying the MD performed for the RV into the CV, in fact, almost always improves the Low Complexity algorithm of the RS (Figures 4.10 to 4.15). Considering that the RV MD copy is computationally free, it must be considered as a very good option for a zero complexity MD in an MVD encoding environment.

The RV MD, given the good correlation with the CV motion field, checked in these experiments, is used to drive the DFMD evaluating the thresholds needed to know which depth differences correspond to motion differences and which, instead, correspond to static areas. By observing the performance of the DFMD algorithm, it is possible to notice that the RD performances obtained by the Low Complexity MD and by the RV MD Copy are always improved (Figures 4.10 to 4.15) and that, in some cases as

Mobile (Figure 4.11) and Cafe (Figure 4.15), the RD curves are pretty close to the High Complexity MD one, obtained using the most complex (and the best in term of RD performance) algorithm.

As explained in Section 4.4.1, the DFMD is improved by considering the disparity between RV and CV in order to provide a more accurate threshold to use in the splitting stage of the mode selection (DDFMD algorithm). The more accurate thresholds provide a noticeable improvement in all the experiments (Figures 4.10 to 4.15). In sequences as Mobile (Figure 4.11), Beergarden (Figure 4.13), Book Arrival (Figure 4.14) and Cafe (Figure 4.15), the DDFMD is able to reach the traditional AVC/H.264 based High Complexity MD which performs the best possible mode selection by testing all the available modes. This good quality is obtained with a considerable complexity reduction which is shown and discussed in the following section.

Computational complexity performance evaluation

In the previous section, the RD performance obtained by the two algorithm designed thanks to the study carried on has been compared with three other algorithms. However, another target of the study in this chapter was to provide a mode selection with a considerable complexity reduction.

A theoretical complexity reduction has been evaluated and depicted in the Section 4.5, where the several cases are considered and the number of operations needed has been calculated. In order to verify in a real situation if the complexity is actually reduced, the time needed to encode the test sequences, using the five considered MD algorithms, has been compared.

Figures 4.16 to 4.21 show the evaluated encoding time for the five algorithms and, for each algorithm, for the four considered QPs. It is noticeable that in all the cases the two algorithms implemented in the reference software (Traditional AVC/H.264 High Complexity and Low Complexity) both need the highest computational time to be performed. In the case of the High Complexity, this higher complexity is compensated by a, almost always, better RD performance than any other algorithm. However, for the Low Complexity case, the quality is almost always the worst one, therefore a so high computational load is worthless.

The third method, the copy of the modes used in the RV, has a complexity comparable to the two algorithm developed in this study, although always slightly lower. The reduced complexity is due to the MD which has been previously performed on the RV. However, as shown in the RD performance evaluation, the quality obtained by this method never reaches the two developed methods (DFMD and DDFMD).

Finally, the DFMD and the DDFMD methods show how the complexity reduction, with respect to the Low Complexity and High Complexity MD, is always considerable and, at the same time, it obtains a quality which almost reaches the best one. Comparing the complexity of DFMD and DDFMD, it is possible to notice that they are usually very similar. That is due to the fact that the only difference between the two algorithms is the use of DVs between RC and CV. This DVs are already evaluated so there is no need of any further operation. Slight differences are present in few cases: these differences are due to the thresholds evaluated between the DFMD and DDFMD methods which differently define if the depth is reliably used for the MD or not. The two decisions may lead to a mayor, or minor, usage of the full search.

Figure 4.10: Sequence Akko&Kayo. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.

Figure 4.11: Sequence Mobile. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.

Figure 4.12: Sequence Balloons. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.

Figure 4.13: Sequence Beergarden. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.

Figure 4.14: Sequence Book Arrival. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.

Figure 4.15: Sequence Cafe. Comparison of the PSNR performance of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes.

Figure 4.16: Sequence Akko&Kayo. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.

Figure 4.17: Sequence Mobile. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.

Figure 4.18: Sequence Balloons. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.

Figure 4.19: Sequence Beergarden. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.

Figure 4.20: Sequence Book Arrival. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.

Figure 4.21: Sequence Cafe. Comparison of the encoding time of the DFMD and DDFMD algorithms vs a traditional AVC/H.264 encoder (High and Low Complexity MD) and the copy of the RV selected modes for four QP values.

4.7. Conclusions

In this chapter, a study about the complexity reduction of the mode selection stage, in an AVC/H.264 encoded environment, for MVD sequences is presented.

The study proposes to analyze the depth sequences (provided for 3D rendering purposes) to extract useful information about the motion of the scene, following the idea that homogeneous depth corresponds to homogeneous motion and, vice versa, a depth gap corresponds to a motion inhomogeneity. However, as the correspondence between depth and motion is not always univocal, to know when the depth analysis is not reliable to detect a motion inhomogeneity, other information can be exploited. In this case, the RV, compressed using a traditional AVC/H.264 encoder, is used to create an approximation of the motion field of the CV.

In this study, two methods able to perform a low complexity MD, have been developed. The first one (DFMD) exploits the information described above and it is able to improve the LC MD algorithm implemented in the RS reducing considerably the complexity. The second one (DDFMD) refines the use of the RV using the DVs and, without any computational load added to the DFMD, increases the performance reaching, in some cases, the HC MD algorithm implemented in the RS.

5 Low Complexity Mode Decision and Motion Estimation for Depth encoding

Chapter abstract

As in the previous one, also this chapter focuses on the complexity reduction of video compression in an MVD environment. The main difference is that, this work focuses on the complexity reduction of the encoding of the depth sequences. The depth information in 3DTV and Free Viewpoint Video is used to provide information about the scene geometry in order to generate virtual views where a camera is not physically present. As a consequence, if the quality of the encoded depth is poor, the quality of the virtual views will be affected. However, the presence of the depth sequences forces the encoder to face an additional computational load. This chapter presents a method for the complexity reduction of the depth sequences encoding that maintains the highest possible quality performances obtained by the traditional AVC/H.264. A Low Complexity Mode Decision and Motion Estimation algorithm for the AVC/H.264 based encoding of depth sequences is proposed. The main idea is that, given that a texture sequence and its associated depth represent the same scene from the same point of view, they should have similar motion characteristics. The complexity reduction of the depth encoding, in the proposed algorithm, is obtained by taking advantage of the texture motion information which has been previously processed by a traditional AVC/H.264 encoder. The characteristics, of depth and texture sequences, are analyzed focusing on their similarities and differences and are exploited to design an algorithm able to detect when the motion information of the texture can be useful for the encoding of the corresponding depth sequence. The proposed method is able to achieve the same quality than traditional Full Search MD and ME as defined in AVC/H.264, with a reduction of up to 58% of the computational burden. The quality has been measured by means of the traditional PSNR and, in addition, with the Video Quality Metric (VQM), known as an objective metric highly correlated to the traditional subjective ones.

5.1. Introduction

Free Viewpoint Video (FVV) and 3D Video (3DV) [3], [4] represent the next generation of video paradigms whose goal is the involvement of the observer thanks to the 3D scene reconstruction or the depth perception without stereoscopic glasses or other additional devices [2]. There are several target fields for 3DV and FVV covering a large number of areas like cinema, home theater and video-conference. In all these areas the immersion is provided by representing the scene complying with the depth and by displaying the 3D world showing a high number of viewpoints to freely change the perspective and

to give to the spectator a real world experience [33]. Immersion performance depends on the number of viewpoints which, when high, improves the 3D sensation. However, when increasing the number of cameras, a considerable amount of information must be transmitted or recorded [5]. To overcome this disadvantage the number of viewpoints captured should be limited, bringing a data decrease, but also a worsening of the 3D feeling. The research found the solution to the issue of finding the best trade-off between the number of viewpoints and the perception of the reality by generating synthetic views located where cameras are not present [6].

This virtual view generation needs to be set in a well-defined environment composed by one or more texture sequences and by their corresponding depth sequences. This setting is called Multiview plus Depth (MVD) environment [8] where the depth information is usually represented as a gray scale video sequence which describes the position of the objects in the scene [7]. The virtual sequence is generated by an interpolation of the warped reference textures to the location of the virtual one, using the depth sequences to locate the objects in the 3D space. One of the synthetic view generation methods is called Depth-Image-Based Rendering (DIBR) [9]. The work presented in this chapter focuses on the compression stage of the depth sequences in this MVD environment.

Taking into account their new features, the MVD encoding should be suitably assessed to obtain improvements in terms of performance or complexity reduction. Focusing on the complexity, due to the presence of the depth, the time needed to encode every view is doubled. Given that a depth sequence represents the same scene captured by the traditional texture camera, some aspects of these two videos are similar, as, for example, the motion of the objects. This motion redundancy can be exploited to reduce the computational complexity of the depth encoding. The related work section (Section 5.2) explains how some methods exploit the analogies between texture and depth and use the texture motion information to reduce the complexity of the Mode Decision (MD) and the Motion Estimation (ME) of the depth encoding. However it may be useful to state the differences between the depth video and the associated texture. The depth sequence only provides information about how the objects are far from the camera, therefore, although the objects in the texture and in the depth are the same, a surface parallel to the camera will be represented with the same depth gray level, losing the information related to the texture patterns. Hence some depth characteristics are definitely different from the corresponding textures and, as a consequence, the motion information sharing could generate quality losses [34].

The work presented in this chapter has a double purpose. The first one is to obtain an analysis of the MVD sequences, focusing on the video characteristics strictly related to the compression and which have a direct impact on the motion evaluation. The target of this study is to know when the sharing of motion information between texture and depth is able to ensure a good Rate Distortion (RD) performance [35], or when the video features are different enough to produce a worsening. This study is centered on the video properties which affect directly the MD and ME stages such as temporal and spatial variances. The second purpose is the introduction of a novel Low Complexity MD and ME algorithm, based on AVC/H.264 [36], which, at the stage of the publication of the work, is the first one able to reduce the depth encoding computational burden obtaining the best quality achievable with a traditional AVC/H.264 encoder using Full Search strategies.

The main novelty is the approach used to detect when the texture information is useful to obtain an acceptable prediction for the corresponding depth. The observations made in the analysis are exploited to design an algorithm able to detect when the reuse of the texture motion information over the depth is a reliable solution and when this option is not optimal. The target quality of the method is the RD performance obtained by the Full Search MD and ME, as defined in the traditional AVC/H.264, therefore, the goal is a complexity reduction without quality worsening.

The depth compression quality is usually evaluated through objective metrics, e.g. the well-known Peak-Signal-to-Noise-Ratio (PSNR), applied directly on the depth data or on the synthetic views. However, due to the rendering algorithms, the synthetic views could be affected by errors or artifacts which are not correctly handled by a fully objective metric, therefore it is important to know how the degradation of the synthetic views, caused by the depth compression, is perceived. The application of subjective experiments requires a huge amount of tests and people involved in the video evaluation, so the use of an objective metric is considered more viable. We propose to assess the depth encoding quality evaluation on the synthesized view by means of the Video Quality Metric (VQM) [37], given its good correlation with the quality perceived by a human observer.

The chapter is structured as follows: in Section 5.2 the related work is reviewed, highlighting the novelties of the proposed method. Section 5.3 shows the study carried out comparing the characteristics of texture and depth sequences. The architecture of the proposed Low Complexity MD and ME algorithm is explained in Section 5.4. Section 5.5 analyses the obtained complexity reduction. Results on RD performance and on complexity reduction are presented in Section 5.6. Finally, Section 5.7 describes the conclusions about the work.

5.2. Related Work

As explained above, MVD sequences are composed by texture and depth information. The research work on encoding optimization of this kind of sequences started exploring new approaches in order to take advantage of their new characteristics or to exploit appropriately the additional information available in this environment. The current state of the art related to the depth encoding optimization is reviewed in this section.

Depth video, due to its specific characteristics, should be handled differently from the texture one, in order to adapt the compression to the content. For example, in depth maps, patterns inside objects are lost and edges are the most relevant information because they define the boundaries of the objects in the 3D scene, so the quality of their reconstruction affects the virtual view rendering. The typical block divisions used in standard video coding could not provide an optimal performance, because large homogenous areas could be divided into small blocks and, on the other hand, objects boundaries could be encoded in the same block. Morvan *et al.* [38] proposed a platelet based depth maps encoder which geometrically adapts the area of the blocks according to the boundaries, refining the prediction in the depth areas which are more relevant in view synthesis.

Another target of the research has been the depth encoding RD optimization. The depth information is used to represent the same scene captured by the corresponding texture cameras, therefore some similarities in terms of motion can be found. Daribo *et al.* [39] considered the correlation between the motion of the texture and of the depth and they proposed a MVs sharing algorithm, where the ME is done by a joint motion estimator able to generate common MVs for both texture and depth. Through this method it is possible to improve the RD performance saving bit-stream because, for the encoding of two videos (texture and depth), only the MVs needed to encode one video are used. Seo *et al.* [40] empirically traced how the depth mode selection is different from the texture mode selection, finding that, in depth maps, direct modes are used most often than in textures. Due to this difference, a MV sharing between texture and depth is not always reliable, so Seo *et al.* designed an algorithm able to select the best candidate for depth between Intra modes, Inter modes and the mode used in the texture encoding. When the texture motion information provides a good prediction, the bit saving is obtained by sending only one bit to indicate the reuse of the texture MVs. In the work of Hewage *et al.* the correlation between texture and depth sequences has been exploited to design a concealment algorithm able to improve the performance in case of error in the transmission of the depth information [41].

Figure 5.1: Encoding time comparison between texture and depth for a set of 5 MVD sequences (*Beergarden*, *Book Arrival*, *Cafe*, *Kendo* and *Newspaper*)

Another issue is the complexity reduction of the depth encoding process. Focusing on it, the time needed for the encoding of the depth sequences has been evaluated and compared to the corresponding textures. Figure 5.1 shows a comparison between the average encoding time needed to process texture and depth of one view of a set of an MVD sequences, observing that the time needed to encode texture and depth is comparable. Hence for the encoding of an MVD sequences, due to the presence of a second video, the depth, associated to every texture, the double of time, or of resources, is required for the encoding.

With the complexity reduction purpose, Oh *et al.* [34] introduced the idea of exploiting the similarities, in motion, with the corresponding textures. By extrapolating the texture motion, a good approximation of the depth motion field can be obtained.

To know how the motion data of the depth is similar to the texture ones, Oh *et al.* observed the MVs selected by a traditional AVC/H.264 encoder to compress both texture and depth. In this analysis, the difference in terms of magnitude between the MVs has been measured: when such difference was low, it was possible to approximate the depth motion with the texture motion information. The results of this analysis were highly dependent on the sequences characteristics, hence it was not possible to know *a priori* when to trust the texture motion field. The found solution consisted in taking advantage of the texture modes and MVs, previously selected, to generate the candidates as depth modes and MVs. The algorithm selects the best ones by RD optimization.

Other researchers considered other features of depth maps. Chiang and Chen [42] exploited depth edges to reduce the complexity of the MD stage. Edges are detected with a Sobel operator and they are used to guide the MB segmentation according to the AVC/H.264 modes. The suggested number of candidates is less than the set of modes allowed in AVC/H.264 so, although in some sequence the RD performance gets worse, the complexity always decreases.

Also the work of Zhu *et al.* [43] focused on the depth motion fields features. In this case, depth maps are divided into sub-regions according to edges or moving objects and, depending on the characteristics of every region, different approaches are used to evaluate the mode selection. A reduced number of modes is considered in every region, so, as in the previous cases, the complexity of the MD stage is reduced. The Zhu *et al.* algorithm obtains a considerable complexity reduction (up to 80%), however the quality is highly dependent on the edge detector refinement which, when reduces the complexity, causes quality losses.

At the publication stage, the works listed in this section represented the most relevant research related to the depth map coding where the depth features are analyzed and where the correlation between depth and texture is exploited. Some of them have the aim to improve the RD performance, others have the computational burden reduction purpose. When the goal is the computational complexity reduction, the RD performance is always worse, therefore, until now, the complexity reduction has been paid with a reduced quality of the compressed signal. The algorithm presented in this chapter has the goal of the complexity reduction of the MD and ME stages in depth maps encoding by exploiting the previously evaluated texture motion information with two main novelties.

The first novelty is the *a priori* analysis of the features which have a direct impact on MD and ME. The study focuses on the statistical features of the video which produce differences in the best modes used to encode the depth maps rather than those used to encode the textures. The temporal and the spatial variance of texture and depth sequences is observed to understand the reasons why, in some regions, the use of the texture motion field to encode the depth is not appropriate. Through this approach, a better knowledge of the depth sequence characteristics is obtained.

The second novelty is the algorithm performance target. The other low complexity methods presented in literature have paid the complexity reduction with RD performance losses. The proposed method is able to predict when the reuse of texture modes and MVs is not adequate because it could cause quality drops. If the texture motion

field reuse is not reliable, other procedures are applied in order to approximate to the best RD performance. So the security of reaching the best quality achievable by a traditional AVC/H.264 based encoder is guaranteed, reducing the encoding process complexity.

Summarizing, as the target of the algorithm is the best quality obtained by a traditional AVC/H.264 encoder, the performance, in terms of distortion, is always better than any other low complexity depth encoding algorithm presented in literature. The complexity is reduced only when it does not produce losses.

5.3. MVD features analysis

As said above, the first goal of this study is to state similarities and differences between texture and depth sequences. Such characteristics are analyzed to know which features can be useful for the design of a low complexity depth encoding algorithm.

The study have been made focusing on those statistical features which affect the MD and the ME stages as, for example, the signal variance.

5.3.1. Sequence features involved in MD and ME

In video coding, compression performance is closely related to the sequence variations: small changes are easier to predict and easier to manage than bigger ones. Video signal variations have different nature and they depend on various factors as changes between the values of adjacent pixels in the same frame or changes between the values of pixels from different frames. The variations in the image values can be analyzed by using probability distribution descriptors as the variance, which shows how far a set of values are spread out from each other.

According to the RD theory [44][45], the distortion (evaluated through the Mean Square Error - MSE) of a Gaussian source X , with variance σ_x^2 , can be written as function of its variance and of the bit-rate R needed for its representation (Eq. 5.1):

$$D_x(R) = 2^{-2R} \sigma_x^2 \quad (5.1)$$

Therefore less variance means less distortion of the compressed signal for a fixed rate, or less rate needed for the signal representation with the same distortion. The MD and ME stages aim to reduce as much as possible the signal variation in order to reduce the distortion or the rate, so the signal variance is the feature which mostly affects these stages. Depending on which one of the two stages is observed, different variances have to be considered. In this section, the variances involved in the MD and the ME are observed and highlighted.

The MD stage is divided into two main parts: Intra Mode decision and Inter Mode decision.

Intra modes are used to compress the MB exploiting the spatial redundancy. In this case, the predicted blocks are generated by techniques based on interpolation, extrapolation or means of the pixels values of the same frame located around the area to be

encoded [46] (for further explanation, see Chapter 2). When the Intra compression is applied, the spatial variance is considered in the process and, when it is reduced, the spatial based compression is able to represent the signal with a reduced number of bits and low distortion.

Inter modes are used to compress the MB by dividing it into sub-sections of different sizes and making the prediction of every subsection through pixels of other frames [46]. The region of any other frame which best approximates the area to compress is found in the ME stage and it is chosen in order to minimize the residual to encode (Chapter 2). As this prediction is made through pixels of the other frames, the temporal variance is the main feature affecting this stage. When the temporal variance is reduced, it is possible to obtain a good compression through the Inter prediction.

As temporal and spatial variances are involved in the MD and ME stages, the analysis is assessed over these features. MD and ME are evaluated independently for every MB of every frame, hence we consider a MB-wise variance evaluated in different ways depending on the cases.

For the spatial variance, pixel values changes, with respect to the adjacent ones, are considered. In this case, the variance is evaluated on the luminance values of a 16×16 MB as:

$$\sigma_B^2 = E[(B - \mu_B)^2] \quad (5.2)$$

where B represents the luminance values of a MB and μ_B is the average luminance value for those luminance values represented by B .

For the temporal variance evaluation, the differences between pixels of two consecutive frames without the application of any motion compensation algorithm are evaluated on the luminance values of a 16×16 MB as:

$$\delta_{i_n} = |B_{i_n} - B_{i_{n-1}}| \quad (5.3)$$

where B_{i_n} represents the luminance value of the i pixel of a MB of the frame n and $B_{i_{n-1}}$ the luminance value one pixel of the MB located in the same position in the frame $n - 1$. The temporal variance is evaluated as the variance of the differences δ_n as:

$$\sigma_{\delta_{i_n}}^2 = E[(\delta_{i_n} - \mu_{\delta_{i_n}})^2] \quad (5.4)$$

In the next sub-section the spatial and the temporal variances of a set of MVD sequences, evaluated as stated above, are observed and analyzed in order to obtain a model to follow in the algorithm design.

5.3.2. The variance of MVD sequences

The goal of the proposed algorithm is to take advantage of the similarities between texture and depth for the complexity reduction of the depth encoding. The purpose

of this study is to know when a direct copy of the texture modes and MVs for the depth encoding provides good performance or when it is preferable the use of other strategies. The signal variance affects the video encoder performance, therefore it is necessary to analyze the texture and depth variances focusing on differences and similarities which are useful for the design of the method. The variances analysis is discussed in the following two sub-sections: the first one shows a visual analysis of the variance differences between the texture and the depth of one sequence frame; in the second, to make consistent the considerations made in the first one, the analysis is done over a large number of frames for a set of sequences.

Visual analysis

The analysis starts by observing one texture frame and its corresponding depth map.

Figure 5.2 shows an example of an MVD frame: a texture frame, with the corresponding depth map, of the sequence *Beergarden*. Observing the frames, some analogous characteristics can be perceived: silhouettes and boundaries belong to the same objects, so similar motion should be noticed along the time. Hence it is apparently reasonable the use of the modes and MVs of the texture for the depth encoding.

(a) Texture Frame

(b) Depth map of the Frame represented in (a)

Figure 5.2: Frame #2 of the 5th view of *Beergarden* (a) with its associated depth map (b).

However the low complexity methods based on the sharing of the motion information between texture and depth show how this strategy often provides quality losses. In order to identify in which situations the re-use of the texture information deteriorates the depth compression quality, the statistics features of the sequences have been observed.

Figure 5.3 shows the variance of every texture and depth MB for the frame number 2 of the 5th view of the sequence *Beergarden*.

Figures 5.3.a and 5.3.c show the temporal variances of the texture frame and the depth map, respectively. Within the texture, high temporal variance is present in almost the whole area of the moving objects. In the depth, due to the more homogeneous signal, a high temporal variance is perceived only on the boundaries.

Considering that the objects in the texture and in the depth are the same, the motion information used for the texture compression should well represent the depth motion, except in a situation: depth boundaries in motion have a different temporal variance than texture ones, so, for MBs including boundaries, the motion prediction estimated for the texture may not be reliable to encode the depth. Assuming the previous statement, the texture motion information should provide good performance when used to compress the depth except on the depth boundaries with high temporal variance (depth moving boundaries).

Figures 5.3.b and 5.3.d show, respectively, the MB-wise spatial variances of the texture frame and of its depth. The images indicate the main difference between the two types of video representation: in the depth map all the patterns are lost and the spatial variance results considerably reduced being very different from the texture one. Therefore, the sharing, from texture to depth, of spatial based prediction modes could make worse the compression performance.

Summarizing, according to this first visual analysis on the MB-wise variances, there are two specific situations where the reuse of the texture information seems not ensuring good compression performances:

- Depth boundaries in motion
- Texture compression based on a spatial prediction

Statistical analysis

To check if the considerations made analyzing the variance of one sequence frame are conclusive, the variance of a set of five MVD sequences has been evaluated. Both temporal and spatial variance of texture and depth of every sequence have been considered, evaluated as the mean value of the MB-wise variances of each frame. The comparison has been represented in the Figures 5.4 and 5.5.

Figure 5.4 shows the spatial variances of texture and depth, confirming that the texture has higher spatial variance due to the presence of patterns. A compression based on the spatial variance, in AVC/H.264, is performed by the Intra modes. Due to the mentioned differences, when a spatial based compression is used in the texture encoding, the performance of using the texture mode for the depth encoding is unpredictable.

Figure 5.5 shows a comparison between the temporal variances of texture and depth. Also in this case, the variance is different but the curves show similar trends, although the values are not the same. This trend indicates that the motions of the two sequences have similarities that can be exploited.

The variance values can be analyzed using the Pearson correlation coefficients evaluated between the depth and texture variances:

Figure 5.3: MB-wise temporal and spatial variance for the texture and depth frame #2 of the 5th view of the sequence *Beergarden*.

Figure 5.4: Comparison between texture and depth spatial MB-wise variances evaluated on a set of five MVD sequences (*Beergarden*, *Book Arrival*, *Cafe*, *Kendo*, *Newspaper*)

Sequence	$\rho_{V_{S,T}, V_{S,D}}$	$\rho_{V_{T,T}, V_{T,D}}$
Beergarden	0.1	0.85
Book Arrival	0.67	0.91
Cafe	0.7	0.36
Kendo	-0.21	0.86
Newspaper	0.24	0.94

Table 5.1: Pearson correlation coefficients evaluated on the temporal and spatial variances between texture and depth.

$$\rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (5.5)$$

where the variables X and Y represent the two signals the coefficients are evaluated on.

For the evaluation of the spatial variance correlation coefficients it is used:

$$\rho_{V_{S,T}, V_{S,D}} = \frac{E[(V_{S,T} - \mu_{V_{S,T}})(V_{S,D} - \mu_{V_{S,D}})]}{\sigma_{V_{S,T}} \sigma_{V_{S,D}}} \quad (5.6)$$

where $V_{S,T}$ and $V_{S,D}$ represent the spatial variances of texture and depth, respectively. For the evaluation of the temporal variances correlation coefficients it is used:

$$\rho_{V_{T,T}, V_{T,D}} = \frac{E[(V_{T,T} - \mu_{V_{T,T}})(V_{T,D} - \mu_{V_{T,D}})]}{\sigma_{V_{T,T}} \sigma_{V_{T,D}}} \quad (5.7)$$

where $V_{T,T}$ and $V_{T,D}$ represent the temporal variances of texture and depth, respectively. The results of the coefficient evaluation are shown in Table 5.1.

Almost all the correlation coefficients evaluated on the temporal variance are high enough to confirm that the trends are similar. On the other hand, the coefficients evaluated on the spatial variance are lower, confirming the poor spatial variance similarities between texture and depth. The correlation coefficients evaluated on the sequence *Cafe* do not follow the trend of the other ones. In this case, the spatial variance correlation is higher than the temporal one. To check if the temporal correlation is actually so poor, the correlation coefficients have been evaluated dividing the sequence into four temporal segments corresponding to every second (Table 5.2).

The results show how in the first three seconds the correlation of the temporal variances is high and how, in the last one, the coefficient is almost zero. Some depth artifacts can be noticed in this segment of the sequence and they have been considered as cause of the low correlation.

Concluding, the similar trend noticed for the temporal variance curves show how the re-use of the texture motion may provide good results in the depth encoding. The differences between the curves are due to what was noticed in the temporal variance visual analysis: texture temporal variance, due to the pattern motion, is generally

Figure 5.5: Comparison between texture and depth temporal MB-wise variances evaluated on a set of five MVD sequences (*Beergarden*, *Book Arrival*, *Cafe*, *Kendo*, *Newspaper*)

Cafe sequence second	$\rho_{V_{S,T},V_{S,D}}$	$\rho_{V_{T,T},V_{T,D}}$
1st	0.95	0.94
2nd	0.19	0.8
3th	0.7	0.93
4th	0.48	0.02

Table 5.2: Pearson correlation coefficients evaluated on the temporal and spatial variances between texture and depth for every second of the sequence *Cafe*.

higher than the depth one but, depth motion edges have temporal variance higher than the texture one. Hence, a moving edge is considered a critical case where it is not possible to trust the texture information.

The other big difference observed is in the spatial variance. The depth spatial variance is always uncorrelated to the texture one and this makes unpredictable the performance of the coding information sharing.

Summarizing, in general it is possible to trust in Inter modes and MVs of the texture for the prediction of the depth maps except on the depth boundaries, where temporal variance is unpredictable. Texture Intra modes are never correlated to depth spatial prediction and they are not useful in the proposed algorithm design.

5.4. Low Complexity MD and ME

According to the analysis made in Section 5.3, the Low Complexity MD and ME algorithm (LCMDME) proposed in this chapter exploits the similarities between texture and depth variance to reduce the depth encoding complexity. Texture motion information, due to the different features, can produce an acceptable amount quality loss when is used in the depth encoding, hence this strategy is not always reliable. This section explains how the LCMDME algorithm manages all the possible cases and obtains a complexity reduction maintaining the best performance.

MD and ME are applied independently on every MB, so the algorithm is described for one MB. Figure 5.6 shows all the steps of the LCMDME.

As explained previously, some critical cases make the reuse of the texture information for the depth encoding not always reliable. The cases considered in the proposed algorithm are two specific situations.

One is the presence of a depth moving boundary. As observed in the variance analysis (Section 5.3), moving boundaries entail high temporal and spatial variances, hence, the use of the modes and the MVs of the texture can cause a quality degradation of the encoded depth. The second situation is the presence of an Intra mode in the texture. When a texture MB is encoded as Intra, only the spatial prediction is considered, but spatial variance of the texture is definitely different from the depth. Due to the presence of these critical cases, the algorithm has been divided into two blocks: the first checks, in two steps (Fig. 5.6), whether to trust in the texture motion information;

Figure 5.6: Flowchart of the proposed LCMDME algorithm.

the second block is reached when the texture motion information is reliable and is used to select the modes and MVs candidates in the depth encoding.

Block 1 is divided into two steps: the first, *Step 1*, through a moving edge detector, is able to recognize moving boundaries inside the depth MB. If a moving boundary is found, the algorithm identifies a critical situation. As the flowchart shows, when the MB has a moving edge, a traditional algorithm for the evaluation of the MD and of the ME is needed. The *Depth Moving Edge Detector* design is explained in Section 5.4.1.

When no moving edge is detected the algorithm can reach the *Step 2*, where the mode used to encode the texture MB is observed. As texture and depth have different spatial characteristics, if the mode of the texture MB is Intra, the texture information is considered not reliable and, also in this case, a traditional algorithm is used.

If there is not a moving edge and the texture MB has been encoded as Inter, the *Block 2* is reached. This block represents the core of the algorithm and selects the candidates for the modes and MVs. First, a distinction between Inter Skip mode and Inter no Skip modes is needed: *Step 3* checks if the texture MB has been encoded as Skip or as any other Inter mode. If the Skip mode is detected, there are not texture MVs to consider and the algorithm goes directly to *Step 5*, avoiding the evaluation of all the remaining modes and MVs which are usually considered. Therefore, when the texture MB has been encoded as Skip the algorithm obtains a huge complexity reduction.

When the texture MB is Inter (except Skip) *Step 4* has to be processed. At this stage, the algorithm reads the MVs of the texture MB and evaluates the ME only according to the texture mode and MVs. Also in this case the complexity reduction is considerable, because only one mode and the corresponding MVs are tested, skipping the computation of all the remaining candidates and the search of all possible MVs.

The *Step 5* has the task of testing the Skip and Intra modes. As the depth spatial variance is not predictable by observing the texture, it is convenient to always check Intra modes. Due to the depth frame features also the Skip mode must be always checked. Skip modes are used mainly in static regions. At the surfaces of depth frames are more homogenous and motion regions could be apparently static, there is a different use of Skip modes compared to the texture. Due to these characteristics, *Step 5* is always processed.

Summarizing, *Block 1* is able to detect when the motion information of the view is reliable and, in case of not reliability, apply a traditional ME and MD algorithm. *Block 2* is reached when it is possible to trust in the texture motion information and it is divided into three steps: *Step 3* checks if the texture MB has been encoded as Skip or not, to organize the different candidates; *Step 4* evaluates the Inter modes and the Inter MVs of the texture MB when it has not been encoded as Skip; *Step 5* is always processed and it is in charge of testing Skip and Intra modes.

5.4.1. The Depth Moving Edge Detector (DMED)

In the *Step 1* of the LCMDME algorithm, a *Depth Moving Edge Detector (DMED)* is used for the detection of one of the critical cases which makes not reliable the texture information for the depth encoding. The variance analysis has shown that the presence

of an edge is a critical situation only when it is moving. The *DMED*, described in the *Step 1* of the Figure 5.6, is considered.

The *DMED* works as follows:

- Edge detector: a Sobel edge detector (with $th = 0$) is applied on the MB.
- Motion detector: the difference between the considered MB and the MB located in the same position in the previous frame is evaluated to detect a movement.
- When both situations happen, a MB with moving edges is detected (critical case) and a traditional MD/ME algorithm is considered.

An edge detector is already implemented in AVC/H.264 based encoders for the Intra Mode Decision, hence there is not an additional load for the Sobel evaluation [47].

5.5. Complexity reduction analysis

Section 5.4 described how the algorithm decides, for a MB, one mode and the corresponding MVs. However, depending on the case, the needed number of operations varies. Therefore the complexity reduction for every possible case has been evaluated.

5.5.1. Traditional AVC/H.264 computational burden

In AVC/H.264, the MD and the ME are performed evaluating the distortion generated and the bits needed to represent the signal compressed through various strategies. The distortion can be evaluated through some techniques (MSE, SAD, SSD, etc.) where, for a 16×16 MB, the pixel-wise differences are measured. The computational burden has been calculated computing the number of 16×16 MB distortion evaluations needed for every possible case.

As explained in the previous section, Intra and Skip modes, are always considered as candidates as in a full search encoding algorithm, hence their burden has not been evaluated. In the comparison, they are called D_{Intra} and D_{Skip} . The modes analyzed in this section are the 7 Inter modes (16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , 4×4), except the Skip one.

For the ME stage, the computational burden has been evaluated considering a full search method conducted on a ± 32 pixel window (65×65 different locations) centered on the origin pixel of the considered MB. When the tested mode is the 16×16 one, the candidates are all the 16×16 blocks located inside the search area. In this case, a number of $65 \times 65 = 4225$ possibilities must be considered, therefore 4225 $D_{16 \times 16}$ operations are needed.

Other possible modes are tested in the encoding process. In the 8×16 mode, for example, a MB is divided into two 8×16 blocks. The prediction is performed by evaluating the distortion between the original 8×16 blocks and any candidate in the search area, therefore 4225 $D_{8 \times 16}$ are evaluated for every block and the total number of operation is $4225 \times 2 = 8450$ $D_{8 \times 16}$. As the distortion evaluation is usually a sum of differences and as the sum is a linear operation, it is possible to affirm that $D_{16 \times 16} = 2$

$\times D_{8 \times 16}$, then, the 8×16 mode evaluation is equivalent to $4225 D_{16 \times 16}$ operations. The same considerations are transferable on any other mode, so, as the analyzed modes are 7, in a traditional MD/ME algorithm the number of operations needed for the Inter modes (except Skip) evaluation in a MB is $7 \times 4225 D_{16 \times 16} = 29575 D_{16 \times 16}$.

5.5.2. LCMDME computational burden

The LCMDME algorithm is able to detect if the texture motion information is useful for the depth encoding or not. In case of detecting a critical situation, it is not possible to trust in the proposed method and a traditional algorithm is needed. Consequently, the computational burden evaluation of the proposed algorithm must be divided according to the reliability of the texture motion information.

If it is possible to exploit the texture information, the algorithm can follow two cases. The first occurs when the texture MB has been encoded as Skip: in this case, except the Intra evaluation, which is always needed, the only Inter mode evaluated is the Skip one, and the other $29575 D_{16 \times 16}$ operations are not needed. The second case arises when the texture MB is not Skip but it has been encoded through another Inter mode with the corresponding MVs. In this situation, the evaluated modes are the Intra, the Skip and only the Inter mode used for the texture MB. As the MVs of the texture have been already evaluated, a test about all the candidates on the search area is not needed and only one $D_{16 \times 16}$ operation is necessary to know the distortion generated by using the texture MVs.

On the contrary, when the texture information is not reliable, the LCMDME method evaluates the motion information through MD and ME algorithms commonly used. Hence, in this case, the complexity is the same than the evaluated in the previous section: $29575 D_{16 \times 16}$.

Table 5.3 shows the computational burden required in all the situations. In the worst case, the same number of operations computed by a traditional MD/ME algorithm is needed for the encoding of a depth MB. In the other situations, the number of operations is significantly decreased, ensuring the complexity reduction.

	LCMDME Algorithm	Traditional AVC/H.264 MD/ME Algorithm
Moving Edge or Texture Intra MB	$D_{Intra} + D_{Skip} + 29575 D_{16 \times 16}$	$D_{Intra} + D_{Skip} + 29575 D_{16 \times 16}$
Skip Texture MB	$D_{Intra} + D_{Skip}$	
Inter (Non Skip) Texture MB	$D_{Intra} + D_{Skip} + 1 D_{16 \times 16}$	

Table 5.3: Computational complexity burden required for the MD/ME algorithm.

5.6. Experimental results

This section presents the performance results obtained using the LCMDME algorithm and its comparison with the ones obtained using other MD and ME algorithms. Ini-

tially, the results have been evaluated in terms of RD performance to check if the goal of the quality preservation is reached. Afterwards, to quantify the complexity reduction, the encoding time saved by the proposed algorithm with respect to the traditional AVC/H.264 has been analyzed.

5.6.1. Experimental setup

The effectiveness of the proposed method is highly related to the quality of the depth sequences because, if the shape of the objects is not correctly preserved, the depth regions could be displaced with respect to the texture. In case of inadequate correspondence between the objects in texture and depth, the texture motion information may not well approximate the depth motion, hence the choice of the sequences used for the test is dependent on the depth maps quality. The sequences have been selected discarding those with a depth quality not considered acceptable in the MPEG evaluations [48]. All the remaining sequences have been used. The selected sequences are described in Table 5.4, where the identification number of the left (L) and right (R) views, the resolutions, and the frame rates are specified.

Sequence	Resolution	L View	R View	Fr. Rate
Beergarden	1920×1080	5	6	25
Book Arrival	1024×768	8	10	16.67
Cafe	1920×1080	2	3	30
Kendo	1024×768	1	3	30
Newspaper	1024×768	2	4	30
Mobile	720×540	4	7	25
Pantomime	1280×960	37	39	25

Table 5.4: MVD sequences used for the experiments.

As it has been explained in the Introduction (Section 5.1), the depth is never displayed and it is used to generate the synthetic views. Hence, the best way to evaluate the performance of an AVC/H.264 based coding algorithm designed for depth sequences, is to compare the quality between the synthetic views generated through the encoded depth maps. Synthetic views are created by interpolating texture and depth information that has been previously encoded [9]. Hence, the encoder settings for textures and depths must be detailed.

The common settings of all the encoders have been selected enabling the features considered useful for making conclusive the experiments. Due to the good spatial compression of the depth, Intra modes are always enabled. B modes might be interesting to observe, however, in terms of prediction procedure, they are very similar to the P ones, hence, the effects of the LCMDME method should be proportional on, both, quality and complexity reduction. Furthermore, the embedding of the algorithm in the B prediction process implies a complex implementation. Due to the similarities between P and B modes, and to simplify the algorithm design, the simplest temporal prediction settings (P) has been selected. As future work, it might be interesting to observe the effects also on the B modes.

The encoder settings are listed below:

- JM Reference Software version 18 [32]
- AVC/H.264 Main Profile
- One Group of Pictures (GOP) for every second
- GOP structure: IPPP...
- Enabled Modes: Intra and Inter

The texture sequences used for the generation of all the synthetic views have been encoded using the previous configuration. In addition, in order to test the algorithm in a reliable scenario, the experiments have been set in a fast encoding environment also for the texture encoding. Hence, for the texture ME, the Fast Full Search algorithm, implemented in the JM, has been used.

The depth sequences have been encoded with the proposed LCMDME and with the following set of MD/ME algorithms used for the comparisons:

- High complexity MD / Full Search ME (Full Search H.264 in the results)
- High complexity MD / Enhanced Predictive Zonal Search (EPZS H.264 in the results))
- Zhu *et al.* Fast MD/ME [43] with Sobel edge detector:
 - with threshold 3 (Zhu *th* = 3 in the results)
 - with threshold 10 (Zhu *th* = 10 in the results)

The EPZS algorithm is used in order to compare the LCMDME to a widely used fast ME algorithm. In addition, the comparison is made also with respect to the algorithm of Zhu *et al.* This choice has been made considering such method the most similar to the proposed one, given that it is based on the texture motion and on the use of depth edges to reduce the number of the candidates. In the Zhu *et al.* algorithm, the edge detection threshold varies the number of MBs where a full search is performed. Hence, when the threshold is low, the method improves the performance but the computational burden increases; on the contrary, higher thresholds reduce the complexity but also the compression quality. For the experiments, the Zhu *et al.* method has been performed with two different thresholds in order to test different conditions.

The virtual views have been created using the View Synthesis Reference Software (VSRS) [49] which needs two real views for the synthesis of a virtual view placed between them.

Two RD metrics have been used to assess the quality of the synthesized view. First, the PSNR is used to have an objective performance evaluation. The distortion is evaluated between the synthetic views generated through the original depths and the synthetic views created through the compressed ones.

As the synthetic views often suffer the presence of artifacts, the quality should be also measured with a subjective assessment, but this solution is not always feasible due to the big amount of tests needed for it. Given its good correlation with the quality perceived by a human observer, the Video Quality Metric (VQM) can be used [37]. VQM is designed to measure the perceptual differences in qualities between an original and a processed video. The perceptual differences are represented by values in a range [0,1], where zero means excellent quality compared to the original.

5.6.2. Performance evaluation

The goal of the proposed algorithm is the computational burden reduction of the ME and MD stages in the depth encoding process maintaining the quality obtained by the Full Search ME performed considering all the modes available in the MD. To test if the method achieves the expected goals, the evaluation has been made first measuring the RD curves obtained by encoding a set of MVD sequences and, second, by comparing the encoding times.

For the VQM curves evaluation, the Command Line Video Quality Metric (CVQM) Software Release Version 2.7 [50] has been used. The CVQM software needs at least 4 seconds of video sequence, hence the number of frames in the experiments is always equal to four times the frame rate.

Figures 5.7.a and 5.7.b show, for the *Beergarden* sequence, the PSNR and VQM RD performance of the proposed method and of the comparing MD/ME algorithms (see Section 5.6.1). In the RD curves it is possible to observe how all the methods used for the comparison start reaching, at low rates, the Full Search performance: in this case the coarse quantization causes the main distortion, so the losses due to not optimally performed MD and ME have a minor role.

With the increasing of the rate, the EPZS and the Zhu *et al.* method cause proportional quality losses. The losses caused by the EPZS are due to zonal search which does not consider all the reference window at the ME stage. In the Zhu *et al.* method the situation is different: it has been purposely designed for the depth map encoding and, usually, it is able to improve the EPZS performance. However, the Zhu algorithm has not been designed for an automatic critical case detection, in fact the performance depends on the threshold of the edge detector (high threshold means a minor critical case detection). On the contrary, the proposed algorithm, with the rate increasing, does not cause a loss of quality and the best performance is always reached. This means that the DMED is automatically able to always identify a critical situation and that the use of the texture MVs on the depth encoding is performed only when the algorithm is sure of the good compression quality.

On the other hand, as the LCMDME is a low complexity algorithm, the reduction of the computational burden must be also measured. In order to determine the complexity reduction, the encoding time of the depth compression performed with the proposed algorithm has been evaluated and it has been compared with the encoding time needed to compress the depth sequences with other encoders. As the textures used for the view synthesis are encoded in the same way in all the cases, in this comparison, only the depth encoding time is considered.

Figure 5.7.c shows the encoding time needed for the compression of the two depth sequences used for the view synthesis. It is possible to observe how, in this case, the proposed algorithm is able to reduce almost the same complexity amount than the other algorithms, when comparing to the full search method. This result is highly dependent on the number of MBs where a critical case is present because, to keep quality, the algorithm is forced to perform the Full Search.

Figures from 5.8 to 5.19 show the results for the other sequences, evaluated with the PSNR and the VQM. As for *Beergarden*, the RD performance of the Full Search is

Figure 5.7: Sequence *Beergarden*. RD and Depth encoding time comparison of the LCDMME vs Full Search ME, EPZS ME and Zhu *et al.* ($th. = 3, 10$).

	LCMDME		EPZS		Zhu (th=3)		Zhu (th=10)	
Sequence	Δ PSNR	Δ rate	Δ PSNR	Δ rate	Δ PSNR	Δ rate	Δ PSNR	Δ rate
Beergarden	-0.04 dB	2.5(%)	-0.22 dB	15.1(%)	-0.11 dB	7.1(%)	-0.15 dB	9.6(%)
Book Arrival	-0.03 dB	1.9(%)	-0.06 dB	4.4(%)	-0.12 dB	7.7(%)	-0.15 dB	10.4(%)
Cafe	-0.01 dB	0.3(%)	-0.8 dB	23.7(%)	-0.19 dB	5.1(%)	-0.35 dB	9.7(%)
Kendo	-0.002 dB	0.33(%)	-0.1 dB	5.2(%)	-0.14 dB	7.64(%)	-0.25 dB	14.3(%)
Newspaper	-0.03 dB	1.6(%)	-0.12 dB	6.2(%)	-0.16 dB	9.9(%)	-0.25 dB	14.9(%)
Mobile	-0.007 dB	0.12(%)	-0.93 dB	15.8(%)	-0.26 dB	4.2(%)	-0.96 dB	32.7(%)
Pantomime	-0.003 dB	0.43(%)	-0.06 dB	5.2(%)	-0.12 dB	10.8(%)	-0.16 dB	15(%)

Table 5.5: Bit rate and PSNR difference with the Bjontergaard measure between the Full Search based AVC/H.264 and the compared four methods.

Sequence	<i>Beergarden</i>	<i>Book Arrival</i>	<i>Cafe</i>	<i>Kendo</i>	<i>Newspaper</i>	<i>Mobile</i>	<i>Pantomime</i>
Saved time (%)	58.3	2.3	42.5	40.9	42.3	34.4	53.6

Table 5.6: Average saved depth encoding time of the LCMDME method with respect to the Full Search ME.

almost reached in all the cases by the proposed algorithm and the little differences are negligible. The performance of the other algorithms follows the previous trend, decreasing the quality as the rate increases. The Zhu *et al.* algorithm always obtains worst quality with the increasing of the edge detector threshold.

The RD performance comparison has been summarized in Table 5.5 where the results are shown using the Bjontergaard Δ Bitrate and Δ PSNR measures [51]. Also, through this metric, the difference between the Full Search and the LCMDME is negligible, instead of the other methods which cause losses in terms of both rate and PSNR.

The figures from 5.20 to 5.25 show the depth encoding time and Table 5.6 shows the percentage of average encoding time saved for all the sequences with respect to the Full Search ME method.

The encoding time comparison shows how the proposed LCMDME obtains different performances, in terms of computational reduction, which vary depending on the sequence. The proposed algorithm is able to check if it is able to obtain the best RD performance with the texture motion information. If not, the Full Search is used ensuring the best quality but the computational burden, for a specific MB, is not reduced. Analyzing the limit cases *Beergarden* and *Book Arrival* (the best and the worst, in terms of complexity reduction), the number of MB encoded through the Full Search in the LCMDME algorithm has been evaluated. The Full Search is performed when a moving edge has been detected or when the corresponding texture MB is Intra. Hence, the evaluation focuses on the percentage of texture Intra MBs and MBs where a moving edge is found; Table 5.7 shows the results. As it is expected, the high percentage of moving edges, in *Book Arrival* implies a low reduction of the load because, in order to reach the best quality, the LCMDME forces the Full Search ME and allows all the modes. On the contrary, the presence of critical cases is very low in *Beergarden* and the algorithm is able to considerably reduce the encoding time.

Figure 5.8: Sequence Book Arrival. Comparison of the PSNR performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.9: Sequence Cafe. Comparison of the PSNR performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.10: Sequence Kendo. Comparison of the PSNR performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.11: Sequence Newspaper. Comparison of the PSNR performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.12: Sequence Mobile. Comparison of the PSNR performance of the LCDMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.13: Sequence Pantomime. Comparison of the PSNR performance of the LCDMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.14: Sequence Book Arrival. Comparison of the VQM performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.15: Sequence Cafe. Comparison of the VQM performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.16: Sequence Kendo. Comparison of the VQM performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.17: Sequence Newspaper. Comparison of the VQM performance of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.18: Sequence Mobile. Comparison of the VQM performance of the LCDMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.19: Sequence Pantomime. Comparison of the VQM performance of the LCDMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.20: Sequence Book Arrival. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.21: Sequence Cafe. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.22: Sequence Kendo. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.23: Sequence Newspaper. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.24: Sequence Mobile. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Figure 5.25: Sequence Pantomime. Comparison of depth encoding time (needed for the encoding of the two considered depth views) of the LCMDME algorithm vs a traditional AVC/H.264 encoder (Full search and EPZS) and Zhu *et al.* ($th. = 3, 10$).

Sequence	I Texture MB	Mov. Edge MB	Total
Beergarden	2(%)	5(%)	7(%)
Book Arrival	5(%)	76(%)	81(%)

Table 5.7: Percentage of Full Search MB for the limit cases (*Beergarden* and *Book Arrival*).

5.7. Conclusions

In this chapter, a new low complexity algorithm to perform the ME and the MD stage for the encoding of depth sequences in an FVV environment has been proposed. To the best of the author knowledge, the LCMDME algorithm is the first low complexity algorithm for depth encoding able to obtain the best RD performance achievable by a traditional AVC/H.264 based encoder reducing considerably the computational burden.

The motion characteristics of the MVD sequences have been analyzed to prevent critical situations and to optimize the strategies to obtain the best performances. The obtained results show that it is possible to reach optimum levels of quality saving a considerable amount of computational load and, consequently, encoding time. The RD performances show how the quality obtained by a traditional AVC/H.264 based encoder is always reached and the computational burden is reduced saving up to 58% of the computational time.

6 Depth Video Coding for Free Viewpoint Video Oriented to the Synthetic View Perceptual Quality

Chapter abstract

As mentioned in Chapter 3, the work introduced in the last part of this thesis focuses on the optimization of the MVD sequences encoding oriented to the preservation of the perceptual quality of the synthetic (or rendered) views generated from existing view plus depth videos. In perceptual video coding, the human visual system characteristics are normally exploited to improve the compression efficiency. As depth sequences are never shown, a perceptual video coding assessed over them would be not effective. This chapter introduces the novel idea of perceptual rate-distortion optimisation assessed over the perceptual distortion of the rendered views generated through the encoded depth sequences. The work can be divided into three different ways to apply, on the depth, the perceptual video coding concepts. The first one is a direct application of the just noticeable difference concept on the synthetic view distortion evaluation. The experimental results show the effectiveness of the proposed method, able to obtain a considerable improvement of the rendered view perceptual quality measured by the Peak Signal to Perceptual Noise Ratio. The first method has however certain drawbacks, especially the limited correlation with subjective metrics. The second method has been designed to correct such limitations, considering also the error in the synthetic view pixel positions. The last method is a natural evolution of the second one. Including the just noticeable difference into the rate-distortion equation, the compression efficiency is significantly improved. Both second and third methods overcome the performance obtained by a traditional AVC/H.264 encoder considering objective and subjective related quality metrics.

6.1. Introduction

The incoming advances in Free Viewpoint Video (FVV) [3] and in 3DTV [33] are improving the observer involvement giving the chance to select an arbitrary point of view in a 3D scene.

In FVV, views are recorded by cameras placed all around the subject and, in order to reconstruct the whole 3D scene, synthetic views are generated where a real camera is not physically present. Synthetic views are generated by interpolating the warped reference textures to the location of the virtual one, using depth sequences to locate

objects in the 3D space. One of the most used synthetic view generation methods is the Depth-Image-Based Rendering (DIBR) [9].

In a video coding environment, depth sequences could be normally handled and compressed through a traditional encoding algorithm. However, they have a different purpose than the texture ones, because they are used only as information about the position of the objects in the scene and they are never displayed. Previous research focused on different compression methods able to improve the encoding performance for this specific content. As the main reason of using the depth is the view synthesis, the effects of the compression have to be analyzed on the virtual views. The study of Merkel *et al.* [52] has shown how the depth compression produces rendered views losses and how, introducing specific depth encoding algorithms, it is possible to reduce this effect. As a consequence, the research on the compression of the depth signal has been oriented more toward an optimization over the view synthesis.

Analysing the state of the art corresponding to the same period of the publication of the compression methods described in this chapter, there was no research work about the possibility of considering compression strategies, for the depth signal, that involved the perceptual quality. In order to fill the gap, we included, in the encoding process of the depth signal, one of the main concepts used in perceptual video coding: the Just Noticeable Distortion (JND) applied on the synthetic views generated by the compressed depth video.

Initially the JND has been directly applied on the pixel values of the synthetic views in a simple but effective method based on a novel Rate Distortion Optimization (RDO) process that minimizes the synthetic view perceptual distortion evaluated through the Mean Squared Perceptual Error (MSPE)[17]. Section 6.3 is dedicated to the explanation of the Depth Perceptual Encoder (DPE), method for the compression of the depth, able to reach a considerable improvement in the synthetic view quality evaluated through the Peak Signal to Perceptual Noise Ratio (PSPNR) designed by Chou and Chen [53].

However, the PSPNR is not correlated to subjective quality tests as other metrics and the results obtained through other metrics (as PSNR) have shown quality degradation. Because of this, an additional depth encoding proposal, based on a novel RDO process, is described following the previous DPE method. The proposal is divided into two methods. The first one is a depth encoder in which, in addition to the traditional distortion, also the error in the pixel positions of the synthetic view is considered. A wrong position of the pixels in the virtual view is indeed caused by the compression distortion of the depth. This first method, described in Section 6.4, is called Synthetic View Pixel Displacement based Depth Encoder (SVPD-DE). The second method, called Just Noticeable Pixel Displacement based Depth Perceptual Encoder (JNPD-DPE, Section 6.5), is an improvement of the previous one thanks to the introduction of a weighting system, based on the HVS and modeled by a novel version of the JND concept, adapted to the depth. The JND, as previously stated, is the minimum visibility threshold below which no change can be perceived by a human observer [54], but it is designed for the traditional video. Therefore we also propose to manage the JND applying a sort of translation from the texture to the depth, considering the distortion the human eye can perceive by displacing the pixel position in texture sequences (Section 6.5.1). The results obtained by the two versions of the second proposed depth encoder show an

improvement in terms of quality which has been evaluated through the Video Quality Metric (VQM), which shows a good correlation with the perceived video quality [37] but also, in many cases, an improvement in terms of objective quality metrics as the widely known PSNR.

6.2. Adopted JND model

As mentioned above, the HVS characteristics are considered in the methods presented in this section. In the Chapter 3, an overview of the HVS and of the JND concept has been provided explaining that the human eye characteristics are modelled and exploited to optimize the encoding process according to the perceived quality.

The JND theory, used in traditional 2D video compression, can be entirely moved to the 3D video textures. However, in order to simplify as much as possible the understanding of the methods and their implementation in the software, we decided to exclusively consider the luminance sensitivity of the human eye modeled on the pixel domain. Luminance differences are perceived depending on the surrounding luminance level which makes some differences more visible than others. The JND model has been defined according to the visual sensitivity masking effects which can depend on the spatial, temporal and spectral signal characteristics [55]. The considered JND model is based on the simplified Spatial JND (SJND) model presented in Section 3.4.1. The SJND thresholds have been modeled with a function of luminance contrast and spatial masking, as:

$$P(i, j)_{\text{simplified}} = \max \left(T^L(i, j), T^t(i, j) \right) \quad (6.1)$$

where T^L and T^t are respectively used to estimate the luminance contrast and the spatial masking. The function T^L is used to evaluate the relationship between the visibility threshold and the luminance difference, where L is the average background luminance. The function T^t is used to measure the relationship between the visibility threshold and the neighbourhood luminance. Focusing on the depth encoding, the mentioned SJND model is applied on the virtual view rendered through the original depth maps. The sequence so rendered is used as reference and, applying the SJND model on it, it is possible to know which distortions, produced by the encoder on the depth, are not perceived by a human observer.

6.3. Depth Perceptual Encoder (DPE) design

In the AVC/H.264 based video compression, the sequence encoding is made through several processes based on different prediction strategies. The video frames are divided into MBs and the prediction is applied on every MB. Every possible strategy produces i) a distortion with respect to the original signal and ii) a certain amount of bits, used for the representation. To improve the encoding efficiency, the encoder selects the one with the minimum rate-distortion cost, between all the possible procedures. The

best prediction strategy is chosen through the RDO process [35] and the RD cost is evaluated as:

$$J(QP, M) = D(QP, M) + \lambda_{QP}R(QP, M) \quad (6.2)$$

where J denotes the RD cost; QP is the quantization parameter used to quantize the signal and M is the mode used for the prediction; D indicates the distortion relative to the considered compression and R the rate (in terms of bits) used to represent the compressed stream; and the value λ_{QP} is the Lagrangian multiplier evaluated, for every QP , minimizing the RD cost function, defined in [35]:

$$\lambda_{QP} = 0.85 \cdot 2^{(QP-12)/3} \quad (6.3)$$

A traditional encoding strategy, in this case, would optimize the RD cost according to the reconstructed depth frames quality. However, as mentioned in the introduction, the research has been oriented to a rendered view based RDO. In a perceptual video coding environment this approach is even more effective due to its characteristics which are completely oriented to the human observer. As a consequence, in this novel Depth Perceptual Encoder (DPE), we propose a new perceptual RDO process, where the distortion is evaluated on the rendered view, taking into account the HVS characteristics modeled by the JND concept. The perceptual RD cost is evaluated as:

$$PJ_s(QP, M) = PD_s(QP, M) + \lambda_{QP}R(QP, M) \quad (6.4)$$

where PJ_s denotes the perceptual RD cost evaluated as in the standard approach (Eq. 6.2) but with a different distortion metric PD_s which represents the perceptual distortion measured between the synthetic view rendered through the original depth and the synthetic view rendered through the compressed depth taking into account the HVS sensitivity modeled by the SJND derived from the work of Chou and Chen [53]. A Mean Squared Perceptual Error (MSPE) method has been used for the perceptual distortion evaluation:

$$PD_s(QP, M) = \frac{1}{256} \sum_{i=1}^{16} \sum_{j=1}^{16} \left((|p_r(i, j) - \tilde{p}_r(i, j)| - S_{JND}(i, j))^2 \delta(i, j) \right) \quad (6.5)$$

where $p_r(i, j)$ is the luminance value of the pixel located in the position (i, j) of the virtual MB rendered using the original depth MB and $\tilde{p}_r(i, j)$ is the same value of the virtual view rendered through the depth MB encoded using the mode M and the quantization parameter QP . The value $S_{JND}(i, j)$, defined in the Eq. 3.4 represents the visibility luminance threshold evaluated for pixel $p_r(i, j)$. The function $\delta(i, j)$ is used to force the subtraction to 0 when the distortion does not reach the SJND threshold:

$$\delta(i, j) = \begin{cases} 1, & |p_r(i, j) - \tilde{p}_r(i, j)| \geq S_{JND}(i, j) \\ 0, & |p_r(i, j) - \tilde{p}_r(i, j)| < S_{JND}(i, j) \end{cases} \quad (6.6)$$

The Lagrangian multiplier λ_{QP} is evaluated as in Eq. 6.3, where a traditional RD cost function is considered. In the DPE, the distortion is evaluated as in Eq. 6.5, where the MSE metric is modified in order to take into account the HVS characteristics. In the proposed method, we consider Lagrangian multipliers evaluated in a traditional fashion which, due to the similar characteristics of the used distortion metric with respect to a traditional one, can well approximate the perceptual RD cost function minimization.

In some perceptual video encoders [56], the strategy followed is to change the QP in function of the signal JND characteristics. Normally higher QPs (which mean worst quality) are used in areas less sensitive to errors in order to save bit-rate when the perceptual distortion does not increase. In this work the perceptual RDO process is applied on all the possible modes available in AVC/H.264 and between various QPs which range around a fixed value. The perceptual RDO process decides the optimum QP and mode (QP_{opt} and M_{opt}) which minimize the PJ_S cost:

$$(QP_{opt}, M_{opt}) = \arg \min_{QP_q} \{ \arg \min_{M_m} \{ PJ_S(QP_q, M_m) \} \} \quad (6.7)$$

where QP_q represents all the allowed QPs and M_m represents all the available modes for every QP. The operations needed for the DPE application on a depth frame are depicted in Algorithm 6.3.1.

Algorithm 6.3.1: DPE(*PerceptualRDO*)

```

for each frame f(n)
  for each frame pixel (i, j)
     $S_{JND}(i, j)$  evaluated as in Eq. 3.4
  for each MB of f(n)
    while  $QP_{min} \leq QP_q \leq QP_{max}$ 
      for each available mode  $M_m$ 
         $PJ_s(QP_q, M_m)$  evaluated as in Eq. 6.4
        if  $PJ_s(QP_q, M_q) < \min(PJ_s)$ 
           $QP_{opt} = QP_q, \quad M_{opt} = M_m$ 

```

As shown in the pseudo-code, the algorithm starts by evaluating the S_{JND} for every pixel of a frame $f(n)$. Then, for every MB in the frame, it evaluates the QP_{opt} and M_{opt} by minimizing the perceptual RD cost evaluated on all the QPs in the range and all the available modes.

6.3.1. Experimental results

In this section the performance obtained by the DPE is shown and compared with the performance obtained by a traditional AVC/H.264 based encoder. The section is divided into three parts: in the first one, the used sequences and the coding settings are depicted, in the second part, the perceptual performance assessment methodology is proposed and the experimental results are provided in the third one.

Experimental setup

As depth maps are never shown in FVV, the performance of a depth coding algorithm has to be evaluated on the virtual sequence rendered through the compressed depth. Three MVD sequences have been used for the experiments. In order to test the algorithm in different situations, the sequences have three different resolutions and one of them has been captured by moving cameras. Table 6.1 shows the used sequences with the corresponding resolution, frame-rate, real camera used and virtual camera rendered.

The DPE algorithm has been implemented on the JM Reference Software (RS) version 18 [32]. The traditional AVC/H.264 based depth coding has been performed on the same RS version without any change. In both cases the encoder settings are depicted as follows:

- JM Reference Software version 18 [32]
- AVC/H.264 Main Profile
- QP varying in a ± 1 range around a fixed QP value
- GOP structure: IPPP...
- Video length: 1 second

Sequence	Resolution	fps	Orig. View	Virtual View
<i>Mobile</i>	720×540	25	4	4.5
<i>Beergarden</i>	1920×1080	25	5	5.5
<i>Kendo</i>	1024×768	30	1	2

Table 6.1: MVD sequences and settings used for the experiments.

For the virtual view rendering, the algorithm proposed by Fehn in [9] has been used. The same algorithm has been implemented in the DPE to obtain the synthetic view perceptual distortion $PD_s(QP, M)$. In the synthetic sequences there are some occluded areas that, warping to the location of a virtual view, can cause the appearance of some de-occlusions (areas with no available color information). To provide good visual results, the rendering algorithms are followed by post-processing techniques for the de-occluded areas filling. In the rendering method implemented in the proposed algorithm, the rendered view de-occlusions are important areas where it is easier to perceptually notice a pixel displacement due to depth compression errors, hence, at this stage of the study, the perceptual distortion (Eq. 6.5) have been evaluated on the rendered frame without considering any filling. However, as the experimental results have to show the perceptual quality of the sequences perceived by the observer, the performance has been evaluated on the rendered views after a simple de-occlusion filling made by repeating the values of the nearest available pixels with the higher depth value [57]. There are not limitations about the use of other filling techniques.

Performance indicators

This work has been conceived for the depth encoding optimization, with the goal to improve the perceptual quality of the rendered views. For the evaluation, a perceptual

quality metric, called Peak Signal-to-Perceptible-Noise Ratio (PSPNR), proposed in [53], is used. The PSPNR is evaluated as:

$$PSPNR = 10 \log_{10} \frac{255^2}{MSPE} \quad (6.8)$$

where $MSPE$ is the Mean Squared Perceptual Error already used in Eq. 6.5 for the perceptual distortion evaluation:

$$MSPE = \frac{1}{256} \sum_{i=1}^{16} \sum_{j=1}^{16} |p_r(i, j) - \tilde{p}_r(i, j)| - S_{JND}(i, j)^2 \delta(i, j) \quad (6.9)$$

The variables used in Eq. 6.9 are the same already depicted for Eq.6.5 and 6.6.

RD performance evaluation

In this section, the performance evaluation measured by means of the PSPNR is shown. The curves are obtained measuring the quality of the rendered virtual view, first through the depth maps encoded with the proposed DPE algorithm and, second, with a traditional AVC/H.264 based encoder.

Figure 6.1 shows, in terms of PSPNR and bit-rate, the gain the DPE algorithm obtains in perceptual performance with respect to the traditional AVC/H.264 encoder for all the considered MVD sequences.

The results of the traditional AVC/H.264 encoders are optimized on the traditional MSE evaluated on the depth maps, therefore the quality of the rendering and of the de-occlusion filling is unpredictable. This is the reason why the perceptual RD curves, in this case, do not follow the typical logarithmic trend. On the contrary, observing the curves obtained using the DPE, the logarithmic trend is better approximated thanks to the perceptual RDO oriented compression. The approximation is not yet perfect due to the presence of a de-occlusion filling algorithm at the post-processing stage which can change the results predicted by the encoder. A logarithmic-like approximation should be obtained with the implementation of a de-occlusion filling algorithm inside the DPE.

The improvement obtained by the DPE is noticeably high in all the cases for all the considered bit rates: 3.5 dB for very low bit rates, 6 dB for bit rates around 1 Mb/s and up to 9 dB for higher rates. These results indicate that the depth maps encoding could be set in a coding environment different to the existing one and oriented to the real function they have, which is the virtual view generation. Virtual views are often affected by the presence of artifacts so the use of a perceptually oriented encoding of the depth maps should be a better option for the depth compression.

Figure 6.1: Comparison of the perceptual performance of the DPE vs. a traditional AVC/H.264 encoder for the sequences *Mobile*, *Beergarden* and *Kendo*.

6.4. Synthetic View Pixel Displacement based Depth Encoder (SVPD-DE)

The method introduced in the previous section (DPE) has however clear limitations due to the fact that the RD optimisation is performed and optimized according to the PSNR [53] which is not correlated to subjective quality as much as other metrics. In addition, the DPE encoder knows *a priori* the virtual view to generate, so the algorithm obtains good results only for that view and it is not able to guarantee an improvement for all the other possible virtual views to generate. Nevertheless, with the method presented in this section, in addition to the traditional distortion, we consider also the error in the pixel positions of the synthetic view is also considered, which is the main effect that the depth encoding distortion can produce on it.

In view synthesis, depth sequences are used as knowledge of how far are the objects from the camera, so the DIBR algorithm knows where to shift the virtual view pixels. When a depth sequence is encoded, a distortion is added to the depth values which can produce errors in the virtual view pixel position. We consider also this distortion in the RDO process defining a Synthetic View Pixel Displacement based Depth Encoder (SVPD-DE).

In traditional video coding the distortion is evaluated only between the original pixel values and the reconstructed ones, using different metrics as, for example, the Mean Squared Error (MSE).

In depth encoding, the pixels contain the depth values of the 3D scene. Therefore, considering d the depth value and $\tilde{d}$ the reconstructed one after the compression, the rendered pixel positions, P_{synth} , are function of the coordinates (x, y) of the texture image pixel, $P(x, y)$, and of the d values:

$$P_{synth} = f(x, y, d), \quad \tilde{P}_{synth} = f(x, y, \tilde{d}) \quad (6.10)$$

Our goal is to evaluate the error between P_{synth} and $\tilde{P}_{synth}$. However, when a depth sequence is encoded, it is not possible to know, *a priori*, where the virtual views generated will be placed. When two real sequences are used for the view synthesis, the virtual ones are placed in any location between them. In this section we set a scenario with two views ($V1$ and $V2$) and we consider the highest pixel displacement case: when encoding the depth corresponding to one texture sequence ($V1$), the synthetic view generated is in the same location as the other real one ($V2$).

The displacement, or shift, with respect the original view has been evaluated as Euclidean distance for both positions:

$$s = d(P_{synth}, P), \quad \tilde{s} = d(\tilde{P}_{synth}, \tilde{P}) \quad (6.11)$$

where s is the shift given by the original depth value and $\tilde{s}$ is given by the reconstructed one.

Chapter 6. Depth Video Coding for Free Viewpoint Video Oriented to the Synthetic View Perceptual Quality

In AVC/H.264, the RDO process is designed on values represented in a $[0, 255]$ range. Hence, the values s and $\tilde{s}$ have been scaled to the same range. The maximum and minimum displacement, s_{max} and s_{min} , corresponding to the maximum and minimum depth values defined for the MVD sequences, are respectively normalized to 255 and 0. From it we map all the other possible displacements values:

$$S = 255 \frac{s - s_{min}}{s_{max} - s_{min}}, \quad \tilde{S} = 255 \frac{\tilde{s} - s_{min}}{s_{max} - s_{min}} \quad (6.12)$$

where S and $\tilde{S}$ are the shift values scaled on the $[0, 255]$ range.

Considering S and $\tilde{S}$, it is possible to evaluate the distortion in terms of the pixel displacement in the synthetic view, therefore compression errors could be voluntarily introduced when the depth distortion does not affect the pixel displacement. However, these voluntarily introduced errors can be propagated because of the temporal prediction of AVC/H.264, producing unrecoverable distortion, so it is important to evaluate also the depth value error. The distortion of the proposed SVPD-DE is evaluated as a weighted function of both displacement and depth errors:

$$D_{SVPD} = \alpha |S - \tilde{S}|^2 + (1 - \alpha) |d - \tilde{d}|^2 \quad (6.13)$$

where α is the weighting factor used to balance the pixel displacement distortion and $(1 - \alpha)$ is the amount of depth distortion allowed to limit the error propagation.

When the distortion is evaluated on the depth values, the RDO evaluates when the produced error is reasonable, given the bit saving with respect to a strategy with best quality but more expensive in terms of bits. This process is not optimized for the pixel displacement distortion, because a pixel in a wrong position could be noticed more easily than a wrong depth value. In an MVD sequence, the bigger the depth differences are in an area, the greater the possibility to notice a synthetic view pixel position error will be. So, we weight the pixel displacement with a value in a $[0, 1]$ range for every MB. The greater is the difference between the maximum and the minimum depth values within the MB, the heavier will be the shift based distortion. Eq. 6.14 shows the evaluation of the parameter α .

$$\alpha = \frac{d_{MB}^{max} - d_{MB}^{min}}{d_{MB}^{max}} = 1 - \frac{d_{MB}^{min}}{d_{MB}^{max}} \quad (6.14)$$

In the SVPD-DE, chrominance values are not considered. The reason for this choice is because the gray level representation of the depth sequences do not need chrominance values.

This proposed RDO method is the basis for the design of the proposed perceptual depth encoding algorithm where the HVS characteristics are considered (Section 6.5.1 and 6.5.2).

6.5. Perceptual compression of depth maps based on Just Noticeable Pixel Displacement

As a natural evolution of the studies introduced in the previous sections, we extend the SVPD-DE has been extended with a weighting system, based on the HVS and modeled by a novel version of the Just Noticeable Distortion (JND) concept, adapted to the depth. The JND is the minimum visibility threshold below which no change can be perceived by a human observer [54], but it is designed for the traditional video. Therefore we also suggest to manage the JND applying a sort of translation from the texture to the depth, considering the distortion the human eye can perceive by displacing the pixel position in texture sequences.

6.5.1. JND adaptation to depth sequences

In perceptual video coding, the HVS characteristics are exploited to reduce the quality when the distortion is not noticed by the human eye. The perceived distortion is modeled by the JND, which represents the threshold below which the errors are not perceived. However, the JND is designed for the traditional video, considering the pixel values perceived by the observer. As the depth sequences are never shown, a novel adaptation of the JND to the depth is introduced: the JND is evaluated for all the pixels of the texture sequence frames and it is translated to the depth domain in the terms of just noticeable pixel position displacement, that we define as Pixel Shift Tolerance (PST).

A virtual view, placed near a real one, is usually a shifted version of the real view with very similar characteristics, so the real view JND should be a good estimation of the virtual view JND.

The pixel-based JND from Yang *et al.* [58] has been taken into consideration. In order to evaluate the PST, the proposed method considers the real view texture values (Y_t) surrounding the pixel (x, y) . The goal is to know when a pixel displacement error can be noticed or not, so it is checked when shifting the pixel $(x \pm i, y \pm j)$ to the position of the pixel (x, y) the difference is perceived by a human observer. To simplify the algorithm we have considered sequences recorded by coplanar cameras with only a horizontal shift:

$$Y_t(x \pm i, y) \tag{6.15}$$

where i is the horizontal difference from the original (x, y) position.

The process starts evaluating if the values $Y_t(x \pm i, y)$ exceed the value $Y_t(x, y)$ more than what is allowed by the JND. If the difference is less, the shift error is not noticed and so the PST is increased by one unit and the next pixels $(x \pm (i+1), y)$ are analyzed. The loop stops when the JND threshold is passed. Algorithm 6.5.1 shows the process.

Algorithm 6.5.1: DEPTH SHIFT TOLERANCE(*DepthJND*)

```

for each textureframe pixel  $(x, y)$ 
  while  $Y_t(x \pm i, y) \leq Y_t(x, y) + JND(x, y)$ 
    and
       $Y_t(x \pm i, y) \geq Y_t(x, y) - JND(x, y)$ 
       $PST(x, y) = PST(x, y) + 1$ 
       $i = i + 1$ 

```

To summarize, the evaluated texture JND is adapted to the depth domain as shift tolerance error in the view synthesis process.

6.5.2. Just Noticeable Pixel Displacement based Depth Perceptual Encoder (JNPD-DPE)

In the previous section (Section 6.5.1) a JND adaptation to the depth domain has been proposed, defining the PST. In this section the design of the Just Noticeable Pixel Displacement based Depth Perceptual Encoder (JNPD-DPE), which exploits the PST, is explained.

The algorithm starts evaluating the rendered pixel positions shift, as in Eq. 6.10, finding the values s and $\tilde{s}$. Given the difference $|s - \tilde{s}|$, the distortion can be evaluated through two possible equations. When the displacement error $|s - \tilde{s}|$ is greater than the PST the error could be noticed by the human eye and the distortion is evaluated as in the first proposed algorithm (Eq. 6.13). When $|s - \tilde{s}|$ is lower than the PST, the pixel position error could be not noticed by the human eye and the distortion can be reduced. So, the terms of Eq. 6.13 are further weighted to take this into account.

Firstly, to simplify, the distortion equation used in the SVPD-DE (Eq. 6.13) is rewritten, dividing it into two distortion terms: the first based on the synthetic view pixel position error (shift error) and the second based on the depth value distortion:

$$D_{SVPD} = D_{shift} + D_{depth} \quad (6.16)$$

where D_{shift} and D_{depth} are defined as:

$$D_{shift} = \alpha |S - \tilde{S}|^2, \quad D_{depth} = (1 - \alpha) |d - \tilde{d}|^2 \quad (6.17)$$

where α is the same coefficient evaluated in Eq. 6.14.

In order to consider the HVS characteristics, Eq. 6.16 is further weighted taking into account the PST.

The distortion given by the pixel position error (D_{shift}) is so reduced by a coefficient β . To maintain the distortion evaluation process compliant with AVC/H.264, the

distortion given by the depth values (D_{depth}) is also reduced by the coefficient $(1 - \beta)$. So the distortion evaluated in the JNPD-DPE is:

$$D_{JNPD} = \beta D_{shift} + (1 - \beta) D_{depth} \quad (6.18)$$

In the virtual view generation there is a maximum allowed displacement produced by the closest objects from the camera (s^{max}). The coefficient β is defined as the proportional difference between PST and s^{max} . So, the closer is the PST to the maximum shift, the lower will be the weight given to the pixel displacement error. When the PST is higher than s^{max} , β is 0 and only the depth distortion is evaluated, reduced by the coefficient $(1 - \alpha)$. When the PST is 0, the coefficient β is 1 and only the pixel position distortion is evaluated, reduced by the coefficient α . The value β is computed as:

$$\beta = \begin{cases} \frac{s^{max} - PST(x,y)}{s^{max}}, & PST(x,y) < s^{max} \\ 0, & PST(x,y) \geq s^{max} \end{cases} \quad (6.19)$$

The proposed JNPD-DPE is so based on a novel RDO process where the distortion is evaluated as depicted in Algorithm 6.5.2.

Algorithm 6.5.2: JNPD-DPE(PST)

```

for each depth MB
  for each depth pixel
    if  $|s - \tilde{s}| \leq PST$ 
       $D = D_{SVPD}$ 
    else
       $D = D_{JNPD}$ 

```

6.6. Experimental results for the SVPD-DE and JNPD-DPE methods

This section show the performance obtained by the two versions of the proposed method that takes into account the error in the pixel position of the synthetic view, compared to the performance obtained by a traditional AVC/H.264 based encoder. First, the experimental setup is described mentioning the sequences, the encoders settings and the metrics used in the test. Then, the improvement obtained will be shown.

6.6.1. Experimental setup

As previously explained, depth is never displayed because is used for the view synthesis, hence the performance is evaluated on the virtual views. For the virtual view generation

Chapter 6. Depth Video Coding for Free Viewpoint Video Oriented to the Synthetic View Perceptual Quality

Sequence	Resolution	fps	L View	R View	V View
<i>Balloons</i>	1024×768	30	1	3	2
<i>Mobile</i>	720×540	25	4	7	5

Table 6.2: MVD sequences and settings used for the experiments.

two views of every MVD sequence have been used. Table 6.2 shows the used sequences specifying resolution and frame rate, the Left and Right views and the Virtual view generated for the performance evaluation.

As the depth is perceptually encoded, the VQM metric has been chosen to assess the quality of the synthesized view given its good correlation with the quality perceived by a human observer [37]. Using the VQM, the perceptual differences are represented by values in the range $[0,1]$, where zero means excellent quality compared to the original. In order to know the algorithm efficiency by the point of view of an objective metric, also the PSNR performance have been evaluated.

As the JND has been modeled on the real texture sequences, a virtual view located in the same position of a real camera is generated, and used as a reference for the quality evaluation. So, the comparison can be made between the synthetic views, generated through the compressed depths, and the real views from the same location. The synthetic views have been generated with the View Synthesis Reference Software (VSRS)[49].

To encode the depth sequences, the Main Profile implemented in the AVC/H.264 Reference Software (RS) version 18 [32] has been used, with the following settings:

- JM Reference Software version 18 [32]
- AVC/H.264 Main Profile
- QP : 22, 27, 32 and 37
- GOP structure: IPPP...
- Intra refresh: 1 second

Inter and Intra modes have been enabled and the High Complexity Mode Decision and the Full Search Motion Estimation evaluated on a search area of ± 32 pixels (default setting of the RS) has been used. The textures used for the view synthesis were the original ones, without any compression.

6.6.2. RD performance evaluation

The performance have been evaluated measuring the quality of the rendered virtual view, through the depth encoded with the two versions of the proposed method (SVPD-DE and JNPD-DPE) and with a traditional AVC/H.264 based encoder.

Figures from 6.2 to 6.5 show the results. Observing the SVPD-DE curves measured by the VQM (Fig. 6.2 and 6.3), there is an improvement which validates the approach of considering the pixel position displacement. The PSNR results evaluated on the SVPD-DE do not show any improvement (Fig. 6.4 and 6.5). This is due to an error

propagation given by the synthetic pixel shift based encoding, because, when the shift quality reduction is not high, the proposed RDO process voluntarily introduces a distortion, in order to reduce the bit-rate. These frames so encoded, are used as references for the encoding of the following ones, so the introduced distortion is propagated and noticed by the PSNR.

Figure 6.2: VQM comparison between the sequences encoded through the proposed algorithms and the traditional AVC/H.264 for the sequence *Balloons*.

Figure 6.3: VQM comparison between the sequences encoded through the proposed algorithms and the traditional AVC/H.264 for the sequence *Mobile*.

The SVPD-DE algorithm is used as baseline algorithm for the JNPD-DPE. Evaluating the performance of this second algorithm, an improvement is noticed in both metrics. The perceptual quality is always improved, as shown in the VQM graphics (Fig. 6.2 and 6.3). The error propagation produced in the SVPD-DE is reduced by the JNPD-DPE

Figure 6.4: PSNR comparison between the sequences encoded through the proposed algorithms and the traditional AVC/H.264 for the sequence *Balloons*.

Figure 6.5: PSNR comparison between the sequences encoded through the proposed algorithms and the traditional AVC/H.264 for the sequence *Mobile*.

which obtains an improvement also in terms of the most common objective metric.

The RD performance comparison has been summarized in Table 6.3 where the results are shown using the Bjontegaard Δ Bitrate and Δ PSNR measurements [51]. It is possible to notice an improvement in terms of PSNR, which reaches 0.3 dB with a corresponding bit saving of 13% for the *Balloons* sequence.

Sequence	VSPD-DE		JNPS-DPE	
	Δ PSNR	Δ rate	Δ PSNR	Δ rate
<i>Balloons</i>	-0.01 dB	+12(%)	+0.3 dB	-13(%)
<i>Mobile</i>	-0.01 dB	+0.5(%)	+0.2 dB	-7(%)

Table 6.3: Bit rate and PSNR difference between the traditional AVC/H.264 and the proposed methods.

6.7. Conclusions

This last chapter introduced a novel approach to reinterpret the perceptual video coding principles and adapt them to be used in an MVD environment. The work has been divided into two parts, starting with an initial work, mainly carried on to test the possibility of the application of the JND concepts on the depth sequences and, afterwards, continued with a more appropriate adaptation of the same concepts to an MVD environment. The results of the study show that, using this novel approach, it is possible to improve the perceptual quality of the virtual views and, in some cases, also the objective quality reaching a quantified improvement of 0.3 dB with a corresponding bit saving of 13% in the best of the obtained results.

However, given the early stage of this work, it is not possible to consider this study concluded. These concepts can, indeed, be improved focusing on the way the JND and the perceptual video coding are used to optimize the compression of MVD sequences. This work can be taken as a starting point for further development to improve the compression of the depth maps toward an optimized perceptual quality of the rendered video.

7 Conclusions and future work

In this thesis, as in any research work, both pros and cons for the suggested methods can be found. This chapter is written stating all the good (positive sides) of the work done, followed, of course, by the bad (negative sides). The “Good” section summarizes the biggest achievements of the several studies, as the reduction of the computational load or the improvement in terms of quality. The “Bad” section wants to highlight how the presented work has some limitations or how certain goals are not reachable if they are outside certain conditions.

This chapter ends with some guidelines about the future work and the possible improvements of the research presented.

7.1. The Good

The goal of this thesis is to suggest novel methods to solve several issues in the compression of the MVD video content such as high complexity, redundancy of operations and lack of methodologies dedicated to the improvement of the quality of the depth.

The thesis can be considered as divided in 3 main research paths: the first and the second focus on the analysis of how certain stages of the compression present a certain amount of redundancy which, if reduced, can provide a considerable reduction of the encoding complexity; the last one focuses on several novel approaches for depth video encoding which process the depth according to the perceptual quality of the rendered video.

7.1.1. MVD encoding complexity reduction

The good, intended as positive sides, of the first and of the second study areas is that, in both cases, a remarkable complexity reduction, up to the 58%, has been reached with a negligible loss in terms of quality. The distortion reduction is actually close to zero in many cases when compared to the most expensive available methods.

The method presented in Chapter 4 exploits the similarities between texture and depth in order to avoid repeating an exhaustive MD for the textures. The depth maps are considered as a segmentation of the scene into objects with different motion characteristics, that was already provided for free. Together with the modes selected for the neighbouring views and together with the disparity information, the depth segmentation is translated into a mode segmentation, respecting the MD rules allowed by AVC/H.264. The complexity reduction is significant in all the cases considered in the experiments, saving up to 40%, and the loss of quality is negligible and, in some cases, even close to zero.

In Chapter 5, the approach can be considered similar to the previous one, with the difference that, this time, texture encoding is being exploited to reduce the computational load of the depth encoding and, also, that is not only the MD being optimized but also the ME. The temporal and the spatial variance of texture and depth are analyzed statistically in order to identify the situations where depth and texture motions are similar enough to copy the encoding parameters of the texture on the depth. The statistical analysis has helped developing an algorithm perfectly able to identify when the texture motion information cannot help the encoding of the depth. The complexity reduction, up to 58%, is then obtained with a quality loss almost always close to zero.

7.1.2. Perceptual Depth Coding

The greatest accomplishment of the perceptual work introduced in this thesis is, for sure, the novelty. No perceptual compression was indeed analyzed before of the publishing of the first perceptual paper presented in Chapter 6. Initially inspired by the work about the depth compression optimised according to the quality of the synthetic views, this thesis introduces a method able to compress the depth content according to the perceptual quality of the synthetic views using the JND threshold. The method has shown improvements in terms of quality using the PSPNR, linked to the perceptual thresholds.

The perceptual depth encoding has been further explored, analyzing the main effect, on a synthetic view, of the distortion introduced by the depth compression: the pixels displacement. Controlling the RD optimization according to the synthetic view pixel position, the performance obtained by the first method gains in terms of consistency and improve the quality also when measured by the traditional PSNR and the VQM.

7.2. The Bad

The good results obtained, and summarized above, certainly come with some limitations. In this section such issues and drawbacks are listed in order to provide an objective analysis of what can be improved.

7.2.1. MVD encoding complexity reduction

Of course, the complexity reduction of a compression algorithm always come with a certain loss in terms of quality. However, in this case, the loss is most of the time negligible, especially considering the big gain in terms of reduction of the computational load. The real limitation of the methods developed can be given by the surrounding environment conditions.

In the case of the work presented in Chapter 4, the limitations are the dependencies with the neighbouring views and with the disparity. The MD of the neighbouring view and the disparity, if not accurate, can provide a wrong segmentation on the depth. If the disparity is too big, also, the scene can be quite different, so also an accurate evaluation may lead to wrong results.

In the case of the work presented in Chapter 5, instead, there is no dependency from other views, so these problems are not an issue. However, the main limitation of this work is the strict statistical analysis of the motion similarities between texture and depth. In order to maximize the quality until reaching the best achievable RD performance, the algorithm is able to identify all the critical situation where a reuse of the texture motion information on the depth is not indicated. The strict rules are keeping a high quality paying the price of a reduction that may be significantly increased even with a little quality loss.

7.2.2. Perceptual Depth Coding

The main limitation of the first perceptual study presented is that the improvement was only appreciable only when measured by the PSPNR metric, that was calibrated using the JND threshold. Using this method, the improvement was depending exclusively on the goodness of those thresholds. This limitation has been removed by the following work that has focused on the main effect of a distorted depth: the displacement of the pixels in the synthetic views. As usually happens in an MVD environment, when part of the compression relies on inter-views dependencies, the quality always depends on the distance between views and the coplanarity (or not coplanarity) of the physical cameras. The work introduced in the last part of the thesis has obtained the provided results in a controlled environment where the depth has an excellent quality and where the cameras did not have considerable distances between each other. If the conditions are different, the experiments may lead to not so good results.

7.3. Future work

The limitations listed in the previous section can be used as a basis of the future work.

7.3.1. MVD encoding complexity reduction

In the case of the complexity reduction, there may be still room for improvement giving more flexibility to the algorithm reducing even more the computational load with a controlled quality loss. However, the main possible improvement could be a merging of the strategies between the two works presented in Chapter 4 and Chapter 5. The first one exploits the interview information, while the second one focuses exclusively on the motion correlation between texture and depth of the same view. Repeating the accurate statistical analysis evaluated between texture and depth, showed in Chapter 5, but taking into account also several different views, it may be possible to obtain a deeper knowledge about the motion of the whole three-dimensional scene that, if properly exploited, may lead to a system able to avoid repeating redundant operation in a smarter way.

7.3.2. Perceptual Depth Coding

The perceptual research presented in this thesis has been one of the very first works using this approach to compress depth sequences. It should be taken as a starting point to explore the possibility of improving the depth encoding based on the visual appearance of the synthetic views created using the compressed depth.

The future research can start by the considerations mentioned in the section related to the limitations of the work. The results obtained show that a perceptual approach for the depth encoding can improve the rendered views quality. However, the best results are obtained by the PSPNR metric, which is not the most reliable. Good results are also obtained using the VQM in the updated algorithms, however it may be interesting to find a way to reach, with the VQM, the same performance of the previous one. The first step of this study would be to apply the proposed algorithm to a wider set of sequences to see the results and analyze the differences depending on the quality of the raw depth and the distance and positions of the physical cameras. Based on those results, the considerations will lead to more accurate ways to define the suggested RD optimization method based on the pixel displacement.

Another consideration can be done about the JND coefficients used, that are exclusively calculated in the pixel domain. The perceptual research has shown that the best perceptual coefficients are the one evaluated on the frequency domain. It would then be interesting to see how a use of the frequency based JND threshold would perform in a perceptually optimized depth encoding algorithm.

7.4. Some extra considerations

As seen in Section 1.3, the research work presented in this thesis has been published between 2008 and 2013.

After the publication of the last paper, the research scene has been evolving together with the technological approach to the 3D video representation. The studies on compression of MVD have evolved as well towards methods exploiting the correlation between textures and depth as the Joint Multiview Video plus Depth Coding (JMVDC), published in 2018 by Veena *et al.* [59], and towards further optimization of the encoding process exploiting machine learning techniques, as in the work of Saldanha *et al.* [60]. In addition, an approach like the MVD representation has been replaced by other 3D images, as 360 video, or volumetric representations, as meshes or point clouds, and the technology behind the MVD, based on traditional cameras plus a depth component, has been replaced by light-field cameras and other 3D sensors able to obtain a higher accuracy, in terms of geometry estimation. The video created is currently used in experiences as Virtual Reality (VR) and Augmented Reality (AR) providing a fully immersive representation, where the user is part of the virtual world. Together with the development of the aforementioned new technologies, the compression work has been evolving, focusing in particular on the compression of point clouds, which are the 3D equivalent of the traditional 2D pixels. The research work about the encoding of point clouds is under process of standardization by the Motion Pictures Experts Group

(MPEG), started with the Call for Proposal issued in 2017 [61] and giving birth to the Point Cloud Compression (PCC) MPEG standards [62].

The techniques presented in this thesis have been obviously designed to exploit the characteristics of an MVD scenario, where the textures are traditional 2D videos and the depth components are gray-scale representations. However, most of the theoretical analysis behind the results obtained could be perfectly applied on the point clouds. The point clouds are volumetric voxelized representations of a 3D scanning where every point of a surface has a position in the 3D world and a corresponding color, normally represented as RGB. These two components are called geometry and color and the compression research is focusing on different techniques that handle the two components separately.

It is easy to notice the parallelism between Point Clouds and MVD, where the depth maps were initially compressed independently from the color, and the possibility of exploiting the work proposed to solve the issues that are being faced in the PCC by the point of view of the computational load reduction, given that the motion of the geometry can be exploited to predict changes of the color representation and vice versa. Also in terms of perceptual quality, it is easy to extrapolate the presented approach, where the depth maps were compressed according to the synthetic view perceptual quality. It would be possible, indeed, to focus on techniques able to compress color and geometry according to the volumetric representation that the end user may perceive, saving bit-rate when a higher quality would not be noticed.

7.5. Conclusions

This thesis has shown several approaches to improve the compression of MVD video sequences exploiting the specific characteristics of this environment, as interview redundancy, disparity, depth and texture motion similarities and differences. Both complexity reduction and video quality have been analyzed and improved. In the case of the complexity reduction, the topic has exhaustively analyzed providing two main methods able to significantly reduce the MD and ME computational charge with a quality loss nearly always equal to zero. In the case of the video quality improvement, an initial analysis is given which provides a knowledge about how the virtual view quality may improve when a perceptual approach is used for the compression of depth maps. The initial results can be considered as a promising starting knowing that, with further research, such approach may lead to excellent synthetic views quality.

List of Publications Related to the Thesis

Journal papers

- G. Cernigliaro, F. Jaureguizar, J. Cabrera, N. García, "Low Complexity Mode Decision and Motion Estimation for H.264/AVC Based Depth Maps Encoding in Free Viewpoint Video", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 23, no. 5, pp. 769-783, May 2013. (doi: 10.1109/TCSVT.2012.2223632). (Relation to Chapter 5)

Conference papers

- G. Cernigliaro, F. Jaureguizar, A. Ortega, J. Cabrera, N. García, "Fast mode decision for multiview video coding based on depth maps", *IS&T/SPIE Int. Conf. on Visual Communications and Image Processing, VCIP 2009*, San Jose (CA), USA, SPIE vol. 7257, pp. 72570N-1-10, 18-22 Jan. 2009. (doi: 10.1117/12.806861). (Relation to Chapter 4)
- G. Cernigliaro, F. Jaureguizar, J. Cabrera, N. García, "Fast mode decision for Multiview Video Coding based on scene geometry", *IEEE Int. Conf. on Image Processing, ICIP 2010*, Hong-Kong, China, pp. 3429-3432, 26-29 Sep. 2010. (doi: 10.1109/ICIP.2010.5652874). (Relation to Chapter 4)
- G. Cernigliaro, M. Naccari, F. Jaureguizar, J. Cabrera, F. Pereira, N. García, "A New Fast Motion Estimation and Mode Decision algorithm for H.264 Depth Maps encoding in Free Viewpoint TV", *IEEE Int. Conf. on Image Processing, ICIP 2011*, Brussels, Belgium, pp. 1013-1016, 11-14 Sep. 2011. (doi: 10.1109/ICIP.2011.6115594). (Relation to Chapter 5)
- G. Cernigliaro, M. Naccari, F. Jaureguizar, J. Cabrera, N. García, "Depth Perceptual Video Coding for Free Viewpoint Video Based on H.264/AVC", *Picture Coding Symposium, PCS 2012*, Krakow, Poland, pp. 153-156, 7-9 May 2012. (doi: 10.1109/PCS.2012.6213309). (Relation to Chapter 6)
- G. Cernigliaro, F. Jaureguizar, J. Cabrera, N. García, "Depth Video Coding for Free Viewpoint Video Oriented to the Synthetic View Perceptual Quality", *IEEE Int. Conf. on Image Processing, ICIP 2013*, Melbourne, Australia, pp. 2068-2071, 15-18 Sep. 2013. (doi: 10.1109/ICIP.2013.6738426). (Relation to Chapter 6)

MPEG contributions

- P. Carballeira, J. Cabrera, F. Jaureguizar, G. Cernigliaro, J. Casal, N. García, "Results on 3DVC EE4 for Newspaper", MPEG 95th meeting, contribution M19378, Daegu, South Korea, Jan. 2011.
- P. Carballeira, J. Cabrera, F. Jaureguizar, G. Cernigliaro, J. Casal, N. García, "Results on 3DVC EE4 for Newspaper", contribution M18363, Guangzhou, China, Oct., 2010.
- P. Carballeira, J. Cabrera, F. Jaureguizar, G. Cernigliaro, J. Casal, N. García, "Results on 3DVC EE4 for Newspaper MVD3", contribution M17892, Geneva, Switzerland, Jul. 2010.
- P. Carballeira, J. Cabrera, F. Jaureguizar, G. Cernigliaro, J. Casal, N. García, "Results on 3DVC EE4 for Beergarden MVD2", contribution M17891, Geneva, Switzerland, Jul. 2010.
- P. Carballeira, J. Cabrera, F. Jaureguizar, G. Cernigliaro, J. Casal, N. García, "Results on 3DVC EE4 for Newspaper MVD3", contribution M17602, Dresden, Alemania, Apr. 2010.
- P. Carballeira, J. Cabrera, F. Jaureguizar, G. Cernigliaro, J. Casal, N. García, "Results on 3DVC EE4 for Beergarden MVD2", contribution M17601, Dresden, Germany, Apr. 2010.
- P. Carballeira, J. Cabrera, F. Jaureguizar, G. Cernigliaro, J. Casal, N. García, "Refinement results on 3DVC EE4 for Newspaper MVD3", contribution M17218, Kyoto, Japan, Jan. 2010.
- P. Carballeira, J. Cabrera, F. Jaureguizar, G. Cernigliaro, J. Casal, N. García, "Refinement results on 3DVC EE4 for Beergarden MVD2", contribution M17217, Kyoto, Japan, Jan. 2010.
- P. Carballeira, J. Cabrera, G. Cernigliaro, J. Casal, "3D Video Coding EE4 Results for Newspaper MVD3", contribution M17041, Oct. 2009.
- P. Carballeira, J. Cabrera, G. Cernigliaro, J. Casal, "3D Video Coding EE4 Results for Beergarden MVD2", contribution M17040, Oct. 2009.

Bibliography

- [1] C. Wheatstone, "Contributions to the physiology of vision - part the first - On some remarkable, and hitherto unobserved Phenomena of Binocular Vision," *Signal Processing: Image Communication*, vol. 128, pp. 371–394, 1838.
- [2] J. Konrad and M. Halle, "3D displays and signal processing," *Signal Processing Magazine, IEEE*, vol. 24, pp. 97–111, Nov. 2007.
- [3] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3D video and free viewpoint video - technologies, applications and mpeg standards," in *IEEE International Conference on Multimedia and Expo (ICME-2006)*, pp. 2161–2164, Jul. 2006.
- [4] A. Smolic, "An overview of 3D video and free viewpoint video," *Lecture Notes in Computer Science*, vol. 5702, pp. 1–8, 2009.
- [5] L. Jiang, J. He, N. Zhang, and T. Huang, "An overview of 3D video representation and coding," *3D Research*, vol. 1, pp. 43–47, 2010. 10.1007/3DRes.01(2010)6.
- [6] K. Müller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "View synthesis for advanced 3D video systems," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–11, 2008.
- [7] M. Tanimoto, T. Fujii, and K. Suzuki, "Improvement of depth map estimation and view synthesis," *ISO/IEC JTC1/SC29/WG11 M15090, Antalya, Turkey*, Jan. 2008.
- [8] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *IEEE International Conference on Image Processing (ICIP-2007)*, vol. 1, pp. I-201–I-204, Oct. 2007.
- [9] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV," *SPIE*, vol. 5291, pp. 93–104, 2004.
- [10] ITU-T Rec. H.264 and ISO/IEC 14496-10, "Advanced video coding for generic audiovisual services (MPEG-4 AVC)," Mar. 2010.
- [11] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3DTV - A survey," *IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Multiview Video Coding and 3DTV*, vol. 17, pp. 1606–1621, Nov. 2007.
- [12] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Efficient prediction structures for Multiview Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Multiview Video Coding and 3DTV*, Nov. 2007.

Bibliography

- [13] G. Cernigliaro, F. Jaureguizar, A. Ortega, J. Cabrera, and N. Garcia, “Fast mode decision for multiview video coding based on depth maps,” in *Visual Communications and Image Processing (VCIP-2009)* (M. Rabbani and R. L. Stevenson, eds.), vol. 7257, pp. 72570N–1–10, SPIE, 2009.
- [14] G. Cernigliaro, F. Jaureguizar, J. Cabrera, and N. Garcia, “Fast mode decision for multiview video coding based on scene geometry,” in *IEEE International Conference on Image Processing (ICIP-2010)*, pp. 3429–3432, Sep. 2010.
- [15] G. Cernigliaro, M. Naccari, F. Jaureguizar, J. Cabrera, F. Pereira, and N. Garcia, “A new fast mode decision and motion estimation algorithm for H.264 depth maps encoding in free viewpoint tv,” in *IEEE International Conference on Image Processing (ICIP-2011)*, pp. 1033–1036, Sep. 2011.
- [16] G. Cernigliaro, F. Jaureguizar, J. Cabrera, and N. Garcia, “Low complexity mode decision and motion estimation for H.264/AVC based depth maps encoding in free viewpoint video,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. PP, no. 99, p. 1, 2013.
- [17] G. Cernigliaro, M. Naccari, F. Jaureguizar, J. Cabrera, and N. Garcia, “Depth perceptual video coding for free viewpoint video based on H.264/AVC,” in *Picture Coding Symposium (PCS-2012)*, pp. 153–156, May 2012.
- [18] G. Cernigliaro, M. Naccari, F. Jaureguizar, J. Cabrera, and N. Garcia, “Depth perceptual video coding for free viewpoint video based on H.264/AVC,” in *Picture Coding Symposium (PCS-2012)*, pp. 153–156, May 2012.
- [19] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. West Sussex, UK: John Wiley & Sons, 2003.
- [20] H. Wu and K. Rao, *Digital Video Image Quality and Perceptual Coding*. CRC Press, 2005.
- [21] K. P. Lim, G. Sullivan, and T. Wiegand, “Text description of joint model reference encoding methods and decoding concealment methods, jvt-0079,” *Tech. Rep.*, 2005.
- [22] P. Yin, H.-Y. Tourapis, A. Tourapis, and J. Boyce, “Fast mode decision and motion estimation for jvt/h.264,” in *International Conference on Image Processing (ICIP-2003)*, vol. 3, pp. III–853–6 vol.2, Sep. 2003.
- [23] M. Nieto, L. Salgado, and J. Cabrera, “Fast mode decision on H.264/AVC main profile encoding based on psnr predictions,” in *IEEE International Conference on Image Processing (ICIP-2006)*, pp. 49–52, Oct. 2006.
- [24] L.-F. Ding, P.-K. Tsung, S.-Y. Chien, W.-Y. Chen, and L.-G. Chen, “Computation-free motion estimation with inter-view mode decision for multiview video coding,” in *3DTV Conference*, pp. 1–4, May 2007.
- [25] D.-H. Han and Y.-L. Lee, “Fast mode decision using global disparity vector for multiview video coding,” in *International Conference on Future Generation Communication and Networking Symposia (FGCNS-2008)*, vol. 3, pp. 209–213, Dec. 2008.

-
- [26] Z. Peng, M. Yu, G. Jiang, F. Shao, Y. Zhang, and Y. Yang, "Fast macroblock mode selection algorithm for multiview depth video coding," in *Chinese Optics Letters*, vol. 8(2), pp. 151–154, 2010.
 - [27] W. Zhu, W. Jiang, and Y. Chen, "A fast inter mode decision for multiview video coding," in *International Conference on Information Engineering and Computer Science (ICIECS-2009)*, pp. 1–4, Dec. 2009.
 - [28] L. Shen, Z. Liu, P. An, R. Ma, and Z. Zhang, "Low-complexity mode decision for mvc," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 21, pp. 837–843, Jun. 2011.
 - [29] S.-Y. Lee, K.-M. Shin, and K.-D. Chung, "An object-based mode decision algorithm for multi-view video coding," in *IEEE International Symposium on Multimedia (ISM-2008)*, pp. 74–81, Dec. 2008.
 - [30] G. Cheung, A. Ortega, and T. Sakamoto, "Fast H.264 mode selection using depth information for distributed game viewing," in *Visual Communications and Image Processing (VCIP-2008)* (W. A. Pearlman, J. W. Woods, and L. Lu, eds.), vol. 6822, SPIE, 2008.
 - [31] MPEG, "Motion Pictures Experts Group." url<http://mpeg.chiariglione.org>.
 - [32] Fraunhofer-Institute HHI, "H.264/AVC JM Reference Software (JM 18)," 2011.
 - [33] A. Vetro, S. Yea, and A. Smolic, "Towards a 3D video format for auto-stereoscopic displays," in *SPIE Conference on Applications of Digital Image Processing XXXI*, vol. 7073, pp. 7073–7073–10, 10 2008.
 - [34] H. Oh and Y.-S. Ho, "H.264-based depth map sequence coding using motion information of corresponding texture video," in *Advances in Image and Video Technology* (L.-W. Chang and W.-N. Lie, eds.), vol. 4319 of *Lecture Notes in Computer Science*, pp. 898–907, Springer Berlin / Heidelberg, 2006.
 - [35] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *Signal Processing Magazine, IEEE*, vol. 15, pp. 74–90, Nov. 1998.
 - [36] ITU-T, "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)," 2003.
 - [37] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Trans. on Broadcasting*, vol. 50, pp. 312–322, Sep. 2004.
 - [38] Y. Morvan, P. H. N. de With, and D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," in *Stereoscopic Displays and Virtual Reality Systems XIII*, vol. 6055, pp. 6055 – 6055 – 12, SPIE, 2006.
 - [39] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Motion vector sharing and bitrate allocation for 3D video-plus- depth coding," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, pp. 1–14, 2009.
 - [40] J. Seo, D. Park, H.-C. Wey, S. Lee, and K. Sohn, "Motion information sharing mode for depth video coding," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON 2010)*, pp. 1–4, Jun. 2010.
-

Bibliography

- [41] C. Hewage, S. Worrall, S. Dogan, and A. Kondo, “A novel frame concealment method for depth maps using corresponding colour motion vectors,” in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON 2008)*, pp. 149–152, May 2008.
- [42] P.-T. Chiang and Y.-C. Chen, “Software and hardware design for coding depth map sequence with texture motion information,” in *IEEE International Symposium on Circuits and Systems (ISCAS-2009)*, pp. 1052–1055, May 2009.
- [43] B. Zhu, G. Jiang, Y. Zhang, Z. Peng, and M. Yu, “View synthesis oriented depth map coding algorithm,” *Asia-Pacific Conference on Information Processing*, vol. 2, pp. 104–107, 2009.
- [44] N. Jayant and P. Noll, *Digital Coding of Waveforms*. Prentice-Hall Signal Processing Series, 1984.
- [45] T. Berger, *Rate Distortion Theory - A Mathematical Basis For Data Compression*. Prentice-Hall, 1971.
- [46] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*, ch. 6, pp. 169–183. West Sussex, UK: John Wiley & Sons, 2003.
- [47] K. P. Lim, G. Sullivan, and T. Wiegand, “Text description of joint model reference encoding methods and decoding concealment methods, jvt-0079,” tech. rep., Apr. 2005.
- [48] MPEG2010/N11095, “Description of exploration experiments in 3D video coding,” *ISO/IEC, JTC1/SC29/WG11, Kyoto, Japan*, Jan. 2010.
- [49] D. Tian, P. Lai, P. Lopez, and C. Gomila, “View synthesis techniques for 3D video,” in *Advances in Image and Video Technology*, vol. 7443, SProc. SPIE, 2009.
- [50] Institute for Telecommunication Sciences - Boulder - Colorado, “Video quality metric (vqm) software),” 2009.
- [51] G. Bjontegaard, “Calculation of average PSNR differences between RD-Curves,” *ITU-T Q6/SG16, VCEG-M33*, Apr. 2001.
- [52] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P. de With, and T. Wiegand, “The effects of multiview depth video compression on multiview rendering,” *Signal Processing: Image Communication*, vol. 24, no. 1-2, pp. 73–88, 2009.
- [53] C. Chou and C. Chen, “A perceptually optimized 3-D subband codec for video communication over wireless channels,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, pp. 143–156, Apr. 1996.
- [54] H. R. Wu and K. R. Rao, *Digital video image quality and perceptual coding*. CRC Press, 2010.
- [55] N. Jayant, J. Johnston, and R. Safranek, “Signal compression based on models of human perception,” *Proceedings of the IEEE*, vol. 81, pp. 1385–1422, Oct. 1993.
- [56] C. Sun, H. Wang, H. Li, and T. Kim, “Perceptually adaptive lagrange multiplier for rate-distortion optimization in h.264,” in *Future Generation Communication and Networking (FGCN 2007)*, vol. 1, pp. 459–463, Dec 2007.

- [57] W. Bruls, C. Varekamp, R. Gunnewiek, B. Barenbrug, and A. Bourge, “Enabling introduction of stereoscopic (3D) video: Formats and compression standards,” in *IEEE International Conference on Image Processing (ICIP-2007)*, vol. 1, pp. I–89–I–92, Oct. 2007.
- [58] X. K. Yang, W. S. Ling, Z. K. Lu, E. P. Ong, and S. S. Yao, “Just noticeable distortion model and its applications in video coding,” *Signal Processing: Image Communication*, vol. 20, no. 7, pp. 662–680, 2005.
- [59] G. P. M. Manasa Veena T., Satyanarayana D., “Joint multiview video plus depth coding with saliency approach,” *Lecture Notes in Electrical Engineering*, vol. 471, 2018.
- [60] M. Saldanha, G. Sanchez, C. Marcon, and L. Agostini, “Fast 3d-hevc depth map encoding using machine learning,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2019.
- [61] ISO/IEC JTC1/SC29/WG11 MPEG2014/N16732, “Call for proposals for point cloud compression,” tech. rep., Jan. 2017.
- [62] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, “Emerging mpeg standards for point cloud compression,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, pp. 133–148, Mar. 2019.