# Adaptação e Optimização de Qualidade em Serviços Futuros de Vídeo 3D

**Depth error concealment methods for 3D video over error-prone networks**

Orientador: Salviano Filipe Silva Pinto Soares
Co-orientadores: Pedro António Amado Assunção
Sérgio Manuel Maciel de Faria

Tese submetida à
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
para obtenção do grau de
DOUTOR em Engenharia Eletrotécnica e de Computadores
de acordo com o disposto no
DR - I série - A, Decreto-Lei n.º 74/2006 de 24 de março e
Regulamento de Ciclo de Estudos Conducente ao Grau de Doutor na UTAD
DR, 2ª série - N.º 149 - Regulamento n.º 472/2011 de 4 de agosto de 2011
DR, 2ª série - N.º 244 - Declaração de retificação n.º1957/2011 de 22 de dezembro de 2011 Vila Real,
2016

*Scientific advisors:*

**Salviano Filipe Silva Pinto Soares**

Professor Auxiliar do
Departamento de Engenharias
Escola de Ciências e Tecnologia da Universidade de Trás-os-Montes e Alto Douro

**Pedro António Amado de Assunção**

Professor Coordenador do
Departamento de Engenharia Eletrotécnica
Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

**Sérgio Manuel Maciel de Faria**

Professor Coordenador do
Departamento de Engenharia Eletrotécnica
Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

*Dedicated to my wife,* ***Silene***

*and my parents,* ***Delfim*** *and* ***Irene***

# Abstract

The research described in this thesis addresses the field of 3D image and video communications over error prone networks, using the multiview video-plus-depth format (MVD). In particular, this thesis presents novel contributions for error concealment of depth maps affected by data loss and investigations on their performance based on both objective and subjective quality assessment via view synthesis. Different types of transmission errors are considered in this work to simulate data loss of depth maps at the receiver, resulting in missing blocks, groups of blocks or even whole depth maps. Since in MVD each depth map is associated with either a single view (video+depth) or two adjacent views (MVD), several combinations of corrupted depth maps and associated data, i.e., texture from the same view and/or depth maps from another view, are exploited for concealment. Geometric characteristics of depth maps, such as contours are investigated by exploiting the similarities between depth maps of adjacent views, adjacent maps of the same view, and also the associated texture images. The contributions of this thesis for depth maps error concealment can be classified into three main groups: spatial domain, inter-view domain and temporal domain. In the spatial domain, error concealment methods were developed relying on the contours of the corrupted depth map itself, followed by a weighted spatial interpolation. This concept is the core of the proposed methods, where two approaches emerge to recover lost contours: The first is based on geometric fitting using Bézier curves, while the second exploits the similarities between the depth maps and the associated texture frame contours. Inter-view domain concealment methods are based on the correlation between depth maps and texture images from different views. Such inherent characteristic of these representation formats allow to exploit similarities between texture views to reconstruct the corrupted depth maps. To this end, two techniques were

proposed: the first one is based on the disparity information computed between texture images from adjacent views, while the second technique is based on block matching using warping functions with geometric transforms. In temporal domain techniques, geometric similarity between texture and depth maps is exploited by using the motion information extracted from the corrupted depth map itself and also from the texture images. These temporal techniques are used along with the previous spatial domain techniques, resulting in improved accuracy and better error concealment performance. These methods were further investigated in order to efficiently recover lost descriptions in multiple description coded (MDC) depth maps. As the coarse depth version significantly affects the quality of the resulting synthesised images, the research problem tackled in this topic was focused on efficient concealment of missing descriptions when a single one is lost. The method proposed to recover corrupted depth maps is based on a coarser decoded version, which is recovered by applying the spatial/temporal error concealment techniques to the received description. The negative effects of losing a depth map description are significantly reduced. Another contribution of this thesis is a quality metric, particularly suited to evaluate views synthesised using reconstructed depth maps. A novel perceptually-aware PSNR metric (pPSNR) was devised and validated through subjective evaluation of synthesised images. A novel aspect of this metric is its ability to capture the user preference on the quality of virtual views synthesised using concealed depth maps. This new objective quality metric relies on the regions that are more perceptually relevant to the observer. Despite using the widely known PSNR, it performs a weighted interpolation of the mean square error (MSE) considering regions with different perceptual relevance. The subjective evaluation results reveal high similarity between the proposed metric and the subjective scores obtained from a significant set of observers. Overall, the results confirm that the novel metric devised in this work is suitable to evaluate the quality of virtual views. Synthesis using reconstructed depth maps affected by distortions caused by transmission errors.

**Key Words:** 3D video, Multiview video-plus-depth, Error concealment, depth maps, 3D quality assessment.

.

# Resumo

O trabalho apresentado nesta tese insere-se no âmbito do tema de cancelamento de erros para mapas de profundidade em vídeo multivista com profundidade (MVD). Em particular, são descritas novas técnicas de cancelamento de erros em mapas de profundidade e apresentado um estudo para avaliar como a qualidade desses mapas influencia a síntese de novas vistas virtuais. Para efetuar a simulação de perdas de dados nos mapas de profundidade no recetor, são considerados diferentes tipos de erros de transmissão. Estes erros podem resultar em perdas isoladas de blocos, grupos de blocos ou até mapas de profundidade completos. Tendo em conta que no formato MVD cada mapa de profundidade está associado a uma determinada vista (vídeo+profundidade) ou múltiplas vistas (MVD) poderão existir semelhanças entre os mapas de profundidade corrompidos e as imagens de texturas/mapas de profundidade da própria vista ou de vistas adjacentes. Com o objetivo de atingir uma maior eficácia no cancelamento de erros, estas semelhanças podem ser exploradas. Assim, tendo em conta a informação geométrica, nomeadamente os contornos, exploram-se as semelhanças entre os mapas de profundidade de vistas adjacentes, entre mapas adjacentes na mesma vista e entre mapas de profundidade corrompidos e as imagens de textura associadas.

As novas contribuições descritas nesta tese no âmbito do cancelamento de erros estão classificadas em três grupos: domínio espacial, domínio inter-vista e domínio temporal. No domínio espacial, as técnicas de cancelamento de erros propostas baseiam-se na reconstrução de contornos corrompidos, seguida de uma interpolação ponderada dos valores da profundidade vizinhos. Neste âmbito, foram desenvolvidas duas técnicas para recuperação de contornos perdidos: na primeira a interpolação dos contornos é efetuada usando curvas de Bézier enquanto na segunda técnica são exploradas as semelhanças entre

os contornos dos mapas de profundidade e a imagem de textura associada. Os métodos de cancelamento de erros inter-vista são baseados nas semelhanças entre mapas de profundidade/imagens de textura de vistas adjacentes, obtendo-se informao sobre a semelhança das imagens de textura para reconstruir os mapas de profundidade.
Estas semelhanças são exploradas através de duas técnicas distintas: o primeiro método é baseado na informação de disparidade calculada a partir de um par de imagens de textura entre duas vistas e o segundo é baseado numa técnica de casamento de padrões entre duas imagens de textura de vistas distintas usando transformadas geométricas. Nos métodos de cancelamento de erros que operam no domínio do tempo, a informação de movimento, i.e. vetores de movimento disponíveis, são extraídos do mapa de profundidade corrompido e da imagem de textura. Esta informação de movimento é depois utilizada para auxiliar a reconstrução dos mapas de profundidade corrompidos, obtendo desta forma um cancelamento de erros eficaz.
Estes métodos foram também propostos para desenvolver novas técnicas de cancelamento de erros em mapas de profundidade num ambiente em que so codificados num esquema de múltiplas descrições (MDC). Quando uma das descrições é perdida, ou seja o mapa de profundidade foi apenas parcialmente descodificado, a qualidade das imagens sintetizadas usando essa informação de profundidade é severamente afetada. O método desenvolvido baseia-se na informação geométrica que se encontra disponível nas descrições que foram corretamente descodificadas e nos valores de profundidade que foram corretamente descodificados nas regiões vizinhas, tanto no domínio temporal como espacial. Deste modo, os efeitos negativos resultantes da perda de uma das descrições pertencentes aos mapas de profundidade são significantemente reduzidos. Foram também efetuadas contribuições no âmbito da avaliação de qualidade de vídeo sintetizado usando o formato MVD. A métrica objetiva de avaliação proposta pretende colmatar a falta de soluções existentes que permitam avaliar mapas de profundidade corrompidos que tenham sido posteriormente recuperados, resultante de erros de transmissão. A métrica proposta (pPSNR) é baseada no PSNR mas tendo em conta as regiões percetualmente mais relevantes, nomeadamente os contornos dos mapas de profundidade onde existem transições mais abruptas dos valores de profundidade. O cálculo é efetuado de modo que seja dada um peso maior a regiões percetualmente mais relevantes. Os resultados obtidos com a métrica objetiva desenvolvida são coerentes com os testes subjetivos efetuados, permitindo concluir que a métrica pPSNR é adequada para avaliar imagens que foram sintetizadas usando mapas de profundidade recuperados.

**Palavras-chave:** Vídeo-3D, MVD, cancelamento de erros, mapas de profundidade, avaliação de qualidade para vídeo 3D.

# Acknowledgements

Firstly, I would like to express my gratitude to my advisors, Dr. Sérgio Faria, Dr. Pedro Assunção and Dr. Salviano Soares for the tireless support of my PhD studies and related research, for their motivation, patience, and immense knowledge. Their guidance was essential to become this thesis possible and made this work a rewarding experience.
I would also like to thank all my colleagues at Instituto de Telecomunicações-Leiria Laboratory, namely Luís Lucas, Lino Ferreira, Pedro Correia, Ricardo Monteiro, João Carreira, Nelson Francisco, André Guarda and João Santos. Their support, opinions about the work that made this thesis possible was very helpful and fruitful. Their good mood and constant willingness to help each other was also a huge bonus that made all this work much more enjoyable.
I would like to acknowledge the financial support provided by Fundação para a Ciência e a Tecnologia (FCT) and Instituto de Telecomunicações. Also a special thanks to the Leiria branch of Instituto de Telecomunicações for providing me such good working conditions in the laboratory.
In my personal life, many people have also contributed to make this work possible, even if this help was not of a technical nature. My parents, Delfim and Irene that always supported and gave me strength throughout my entire life who are the reason for the person I am today. At last, but surely no least, I sincerely would like to thank to my wife Silene, for always being there for me, being the pillar of my life.

UTAD, Vila Real — Sylvain Tony Antunes Marcelino
02 de Setembro, 2015

# Index

**Abstract** **ix**

**Resumo** **xi**

**Acknowledgements** **xiii**

**List of tables** **xvii**

**List of figures** **xx**

**List of Abbreviations** **xxv**

**1 Introduction** **1**
1.1 Context and motivation . . . . . 1
1.2 Goals and original contributions of this thesis . . . . . 3
1.2.1 Spatial and inter-view depth map error concealment . . . . . 4
1.2.2 Temporal and MDC error concealment techniques for depth maps . 6
1.2.3 Quality assessment of error concealed depth maps for MVD . . . . 7
1.3 Thesis structure . . . . . 8

**2 3D/Multiview formats and coding methods** **11**
2.1 3D displays . . . . . 11
2.1.1 Stereoscopic displays . . . . . 11
2.2 3D formats . . . . . 16
2.2.1 Multiview Video . . . . . 16

2.2.2 Multiview Video-plus-Depth . . . 18
2.2.3 Layered Depth Video . . . 20
2.3 3D/Multiview Coding methods . . . 21
2.3.1 Simulcast . . . 22
2.3.2 Stereo interleaving . . . 23
2.3.3 Multiview Video Coding . . . 24
2.3.4 Multiview Video-plus-Depth coding . . . 26

**3 Error concealment techniques 33**
3.1 Overview . . . 33
3.2 Spatial Error Concealment for 2D video . . . 34
3.2.1 EC-Spatial domain interpolation . . . 35
3.2.2 EC-Frequency domain interpolation . . . 41
3.3 Temporal Error concealment . . . 45
3.3.1 Frame Copy (FC) . . . 45
3.3.2 Motion Copy (MoCp) . . . 46
3.3.3 Motion Boundary Match Algorithms (MBMA) . . . 47
3.3.4 Motion Vector Extrapolation (MVE) . . . 54
3.4 Error concealment for 3D and multiview-video . . . 64
3.5 Depth map error concealment for MVD . . . 74

**4 Spatial and inter-view error concealment for depth maps 85**
4.1 Depth map error concealment using Bézier curve fitting . . . 85
4.1.1 Edge extraction with a variance method . . . 86
4.1.2 Matching end-points . . . 86
4.1.3 Contour reconstruction with Bézier curves . . . 88
4.1.4 Depth map reconstruction . . . 90
4.1.5 Experimental results . . . 92
4.2 Depth map error concealment using texture image contours . . . 95
4.2.1 Exploiting similarities between depth maps and texture . . . 95
4.2.2 Contour extraction - Canny edge detection . . . 96
4.2.3 Contour reconstruction . . . 97
4.2.4 Depth map reconstruction . . . 99
4.2.5 Experimental results . . . 99
4.3 Depth map error concealment using combined contour reconstruction with Bézier curves and texture information . . . 103
4.3.1 Experimental results . . . 103
4.4 Depth map error concealment using disparity information . . . 106
4.4.1 Stage 1 - Depth recovery using disparity compensation . . . 108
4.4.2 Stage 2: Depth contour reconstruction . . . 111

4.4.3 Experimental results . . . 112
4.5 Depth map error concealment for using BMGT . . . 120
4.5.1 Block Matching with Geometric Transforms . . . 121
4.5.2 Fast search with BMGT . . . 125
4.5.3 Error concealment using BMGT . . . 126
4.5.4 Experimental Results . . . 128

**5 Error concealment for multiview video-plus-depth 133**
5.1 Inter-view and inter-frame error concealment using BMGT . . . 133
5.1.1 Error concealment using motion from texture . . . 135
5.1.2 Depth error concealment using BMGT . . . 135
5.1.3 Spatial recovery of missing depth . . . 137
5.1.4 Simulation results and discussion . . . 138
5.2 Depth error concealment for multiple description decoders . . . 144
5.2.1 Multiple description coding of depth maps . . . 145
5.2.2 Enhancement and error concealment of MDC slice based depth maps - Intra . . . 146
5.2.3 Experimental results . . . 149
5.3 Enhancement and error concealment of MDC slice based depth maps - Intra and Inter techniques . . . 153
5.3.1 Temporal reconstruction . . . 154
5.3.2 Spatial interpolation . . . 155
5.3.3 Experimental results . . . 156

**6 Perceptually-aware quality metric for performance evaluation of depth error concealment 161**
6.1 Quality evaluation of synthesised views . . . 161
6.1.1 Methods used for performance evaluation . . . 162
6.2 Perceptually-aware objective metric . . . 163
6.3 Experimental setup . . . 165
6.3.1 Subjective Evaluation . . . 166
6.3.2 Objective Evaluation *vs.* Subjective evaluation. . . . 169

**7 Conclusions and future work 175**
7.1 Future work . . . 178

**References 179**

**A Test sequences 197**

# List of Tables


2.1 MV-HEVC VS. SIMULCAST: Bit-rate reduction [47]. . . . . 26

4.1 Tested sequences (dB). . . . . 92
4.2 Results for the synthesised images (dB). . . . . 94
4.3 Experimental image synthesis results (PSNR(dB) and SSIM[0-100]). . . . . 101
4.4 Experimental image synthesis results (PSNR(dB)). . . . . 106
4.5 PSNR (dB) of synthesised views using the recovered depth maps under three different packetization modes and four PLR. . . . . 117
4.6 PSNR (dB) of synthesised views using the recovered depth maps in the case of full frame loss. . . . . 117
4.7 Used images corresponding with the respective cameras. . . . . 128
4.8 Scenario 1: Experimental image synthesis results. . . . . 129

5.1 Tested video sequences. . . . . 137
5.2 Mean PSNR-Y(dB) of synthesised views using recovered depth maps. . . . 141
5.3 Index assignment matrix ,$k$=1 . . . . . 146
5.4 Average PSNR-Y of synthesised views . . . . . 151
5.5 Average PSNR-Y of the synthesised views. . . . . 158

6.1 Used images and the corresponding characteristics. . . . . 165
6.2 Test cases for the subjective assessment . . . . . 167

6.3 Relation between objective and subjective metric. . . . . 172
6.4 Synthesised views objective evaluation using PSNR(dB), SSIM(%) and pP-SNR(dB). . . . . 173

# List of Figures

1.1 Pixel and number of views evolution in a recent past [4]. . . . . 2

2.1 Anaglyph glasses. . . . . 12
2.2 Example of an anaglyph image. . . . . 13
2.3 Circular polarization example [24]. . . . . 14
2.4 Auto-stereoscopic displays. . . . . 16
2.5 Example of different camera arrangements [27]. . . . . 17
2.6 MVD: Video image and corresponding depth map (Breakdancers sequence) [30]. . . . . 19
2.7 Example of LDV images and respective residue [34]. . . . . 21
2.8 Example of a simulcast coding scheme [37]. . . . . 22
2.9 Three examples of stereo interleaving [45, 46]. . . . . 23
2.10 Multiview Video Coding (MVC) structure [37]. . . . . 24
2.11 Rate-distortion example for several test sequences encoded with MVC (extension of the H.264/AVC): Ballroom and RaceFigure 2.11 [7]. . . . . 25
2.12 Extensions of the current coding standards (H.264/AVC and H.265/HEVC)for MVD [56]. . . . . 27
2.13 Partition-Based Depth Intra Coding and the corresponding partition pattern [69]. . . . . 30

3.1 Spatial interpolation using weighted interpolation. . . . . 36
3.2 Aplying the *Sobel* mask in the lost region. . . . . 37
3.3 Example of contour edge endpoints matching and linking. . . . . 39
3.4 Six different cases for weighted interpolation pixel selection[99]. . . . . 40
3.5 H.264/AVC intra prediction directions. . . . . 41
3.6 DCT coefficients categorization for a $8 \times 8$ block. . . . . 43
3.7 Missing region and respective support area [118]. . . . . 44
3.8 Example of Motion Copy(MoCp) EC method. . . . . 46
3.9 Example of a typical boundary matching process. . . . . 48
3.10 Concealment using $4 \times 4$ neighbours blocks [127]. . . . . 51
3.11 Motion vector extrapolation [134]. . . . . 55
3.12 Generation of the MV history [135]. . . . . 57
3.13 Pixel based Motion Vector Extrapolation example (pMVE) [137]. . . . . 59
3.14 Example of Hybrid Motion Vector Extrapolation (HMVE) [138]. . . . . 60
3.15 Bi-directional motion copy EC [139]. . . . . 63
3.16 Inter-view and inter-frame correlations [142]. . . . . 65
3.17 Detection of occluded region between views [146]. . . . . 68
3.18 Temporal and interview EC model [150]. . . . . 71
3.19 T. Chang's stereoscopic coding scheme [160]. . . . . 77
3.20 C. Hewage texture and depth error concealment [163]. . . . . 80
3.21 X. Zhang disparity MVs computation based on depth maps [166]. . . . . 82

4.1 Depth map error concealment using Bézier curves block diagram. . . . . 86
4.2 Depth map and extracted contour, from first frame of Champagne sequence, 39th camera. . . . . 87
4.3 Match lost contours and Bézier control points. . . . . 87
4.4 Tangent angle computation. . . . . 88
4.5 Weighted interpolation of the depth values of the lost block. . . . . 91
4.6 Recovered Depth Maps (20% lost blocks). . . . . 93
4.7 Synthesized Views (20% lost blocks). . . . . 94
4.8 Proposed method: processing stages. . . . . 96
4.9 Example of extracted contours: Texture image (light gray), depth map (dark gray), overlapped (black). . . . . 98

4.10 Matching texture image contours to depth map. . . . . 99
4.11 Recovered depth maps with the respective synthesised views (20% of block loss). . . . . 102
4.12 Proposed algorithm diagram. . . . . 103
4.13 Depth maps and the respective synthesised views (20% of block loss). . . . 105
4.14 Depth loss cases in spatial MVD, characterised by the data available for error concealment. . . . . 107
4.15 Algorithmic structure of the error concealment method. . . . . 108
4.16 Depth map block matching using disparity. . . . . 109
4.17 Intensity compensation of the recovered depth map. . . . . 112
4.18 Error patterns originated from different packetisation modes. . . . . 114
4.19 Bland-Altman plots: proposed (PM(a)) . . . . . 116
4.20 Example 1: Depth maps and respective synthesised views (example of 2 lost packets using packetization 3 in the 1st frame of *Champagne*), at 20% PLR. . . . . 118
4.21 Example 2: Depth maps and respective synthesised views (example of 2 lost packets using packetization 3 in the 1st frame of *Book Arrival*), at 20% PLR. . . . . 119
4.22 Depth error concealment with BMGT. . . . . 121
4.23 Proposed algorithm diagram with BMGT. . . . . 122
4.24 Example of a grid deformation used in BMGT. . . . . 122
4.25 Grid Interpolation. . . . . 124
4.26 Quadrilateral diamond search. . . . . 126
4.27 Stage 2: Functional Diagram. . . . . 127
4.28 Depth maps with the respective synthesised views (40% of block loss with error pattern 1). . . . . 130

5.1 Block diagram of the proposed depth map error concealment method. . . . 134
5.2 Objective reconstruction quality (PSNR) of synthesised view luminances. . 138
5.3 Example of damaged and recovered depth maps with the corresponding synthesised images, sequence Ballet, PLR=10% using Packetization 2 (Frame 16). . . . . 143
5.4 Enhancement diagram for MDC depth maps. . . . . 147

5.5 Example of using depth contours . . . . . 148
5.6 MDC depth maps and respective synthesised views at 10% of packet loss (frame 84). . . . . 152
5.7 MDC depth map reconstruction algorithm and view synthesis. . . . . 154
5.8 Texture and depth GOP structure. . . . . 155
5.9 Selection of depth values for interpolation. . . . . 156
5.10 MDC depth maps and respective synthesised views at 10% of packet loss (frame 12), Slice mode. . . . . 160

6.1 Corrupted depth map, Book Arrival, PLR of 20%. . . . . 167
6.2 Subjective results obtained from the 36 observers. . . . . 168

A.1 First frame of *Ballet* sequence (view 0). . . . . 200
A.2 First frame of *Breakdancers* sequence (view 0). . . . . 201
A.3 First frame of *Beergarden* sequence (view 1). . . . . 201
A.4 First frame of *Kendo* sequence (view 0). . . . . 202
A.5 First frame of *Balloons* sequence (view 1). . . . . 202
A.6 First frame of *Book Arrival* sequence (view 8). . . . . 203
A.7 First frame of *Champagne* sequence (view 39). . . . . 203
A.8 First frame of *Dancer* sequence (view 1). . . . . 204
A.9 First frame of *Shark* sequence (view 1). . . . . 204
A.10 First frame of *Newspaper* sequence (view 1). . . . . 204

# List of Abbreviations

2D: Two dimensional.
3D: Three dimensional.
3DTV: Three Dimensional Television.
AVC: Advance Video Coding.
AR: Auto Regressive model.
ATM: Asynchronous Transfer Mode.
BMA: Block Matching Algorithm.
BMGT: Block Matching using Geometric (spatial) Transformations.
BDBR: Bjontegaard Delta Bit Rate.
BR: Bit-Rate.
CU: Coding Unit of H.265/HEVC.
DCP: Disparity Coding Prediction.
DCT: Discrete Cosine Transform.
DLT: Depth Look up Table.
DIBR: Digital Image Based Rendering.
DMM: Depth Modeling Modes.
DS: Diamond Search.
DC: Depth Coding.
EC: Error Concealment.
EPZS: Enhanced Predictive Zonal Search.

FC: Frame Copy.
FFT: Fast Fourier Transform.
FMO: Flexible Macroblock Ordering
FTV: Free-viewpoint TV.
HEVC: High Efficiency Video Coding.
HVS: Human Visual System.
IDR: Instantaneous Decoding Refresh
JVT: Joint Video Team.
LDI: Layered Depth Image.
LDV: Layered Depth Video.
MAD: Mean of Absolute Differences.
MB: Fundamental coding processing unit (Macroblock)
MBMA: Motion Boundary Match Algorithms.
MPEG: Moving Pictures Expert Group.
MoCp: Motion Copy.
MMP: Multidimensional Multiscale Parser.
MV: Motion Vector.
MVE: Motion Vector Extrapolation.
MVC: Multiview Video Coding.
MVD: Multiview Video-plus-Depth.
MVV: Multiview Video.
MSE: Mean Square Error.
NAL: Network Abstraction Layer.
PLR: Packet Loss Ratio.
PPS: Picture Parameter Set.
PSNR Peak Signal to Noise Ratio.
QoE: Quality of Experience.
QP: Quantisation Step.
RD: Rate-distortion.
RT: Real time.
SAD: Sum of Absolute Differences.
SDC: Single Description Coding.

SEC: Spatial Error Concealment.

SEI: Supplementary Enhancement Information message.

TEC: Temporal Error Concealment.

TSS: Three Step Search.

SVC: Scalable Video Coding.

VCL: Video coding Layer.

VSO: View Synthesis Optimisation.

VSRS: View Synthesis Reference Software.

# 1 Introduction

## 1.1 Context and motivation

In the last few years, services and applications using 3D images and video have been continuously growing. The use of 3D media content crosses many different fields, reaching an increasing number of consumers in quite diverse activities. Besides entertainment, there are also several other professional fields where 3D images and video applications are becoming quite useful, such as medical, engineering, manufacturing, etc.

In entertainment, such as cinema, gaming and virtual reality, classic stereoscopic 3D images and video are commonly used to provide depth sensation to users. Volumetry, is another concept used for viewing 3D content using 2D or holoscopic displays, enabling the user to navigate through multiple perspective views around visual scenes by providing the enhanced capability of choosing any arbitrary point of view to match them. This form of 3D viewing can be used in several types of applications, such as FTV (Free-viewpoint TV), augmented reality, gaming, etc [1, 2].

In the medical field, 3D imaging technologies are also becoming useful and common in some specialties. For example, these technologies are of great importance for image-guided radiotherapy and radiosurgery, leading to more effective interventions and reducing human impact, which is a very important concern for the well-being of the patients [3].

In the vast field of engineering and manufacturing, 3D vision and modelling find many

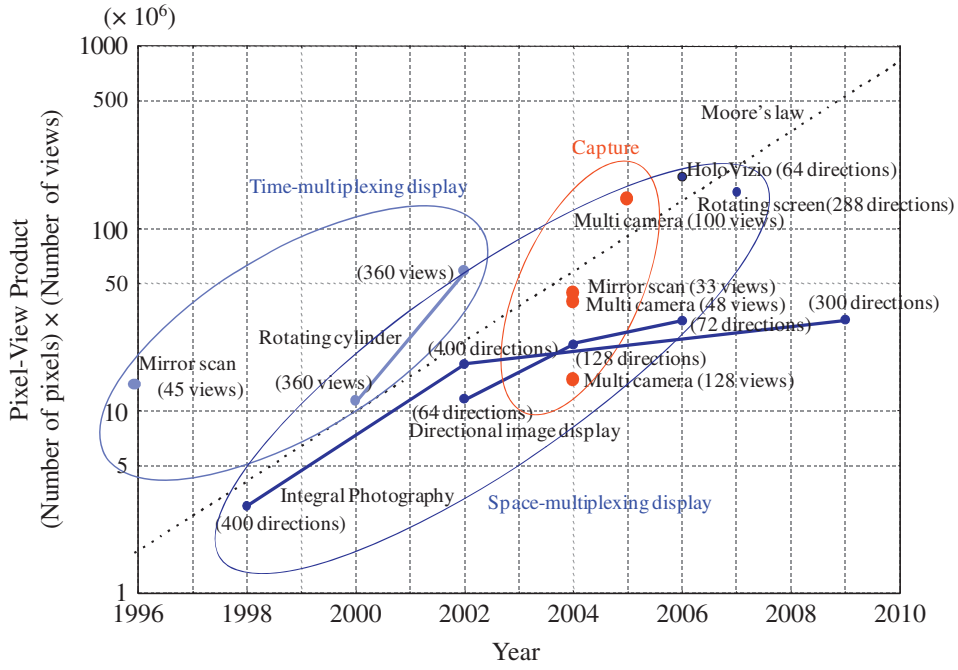



**Figure 1.1** – Pixel and number of views evolution in a recent past [4].

different applications. Yet, for these of applications, accurate depth acquisition and modelling, as well as object tracking remain as challenging tasks that highly influence the effectiveness of 3D computer vision [5].

As a consequence of this recent evolution of 3D technologies across such a large diversity of applications, compression of 3D video and multiview visual content has also been under research and fast development in the last years. In digital multimedia communications, this is expected to lead to a significant increase in the overall multimedia traffic delivered through future media networks. As can be observed in Figure 1.1, the evolution of the spatial resolution (in pixels) over the last few years have been rapidly increasing, both in the 3D content acquisition and display capability [4]. As 3D display technologies become more accessible to the general public, the demand for new and better services also increase, leading to the development of new methods and tools to represent, encode and synthesise 3D content. Nowadays, there are two main coded representations of 3D and multiview video content: Multiview Video Coding (MVC) and Multiview Video-plus-Depth (MVD) [6, 7]. The increasing number of the pixels used in these visual representation formats, particularly those using multiview video, is leading to an enormous amount of data to be

transmitted over digital networks. Since transmission of compressed signals over error-prone channels suffer from errors and data loss, the quality of services and applications using this type of data can be significantly degraded if no specific solutions are implemented to minimise the effect of such problems. It is known that the impact of transmission errors is higher in 3D video communications than in traditional 2D communications, because the quality of experience (QoE) is highly sensitive to a wider variety of quality factors [8]. Therefore, careful design of error concealment algorithms for 3D image and video decoders is essential to minimise the perceived artifacts produced by 3D video data loss. In the particular case of the MVD format, the quality and integrity of the depth map plays a crucial role in the quality of the synthesised views, with great impact on the perceived quality. Hence, the efficiency of error concealment algorithms specifically tailored for depth maps is a critical performance factor in 3D video services [9]. The research described in this thesis is concerned with the problem of recovering lost data in depth maps and its impact on the quality of virtual views generated by using such depth maps.

## 1.2 Goals and original contributions of this thesis

The work described in this thesis is focused on error concealment and restoration of depth maps in MVD format. Novel algorithms are proposed to conceal the effect of various types of errors, including loss of isolated regions and loss of entire depth maps. The proposed techniques include spatial domain error concealment methods, used for intra frame reconstruction, as well as inter-view and inter-frame techniques. Additionally, new enhancement/restoration techniques for depth maps decoded from multiple description coded (MDC) bitstreams were also investigated. The performance through objective and subjective testing quality evaluation of the error concealment algorithms is evaluated in this thesis. An novel objective quality metric is proposed, user-driven features that affect the perceived quality.

The research carried out in the scope of this thesis resulted in several original scientific contributions that can be categorised in the following types: spatial and inter-view depth map error concealment, temporal depth map error concealment and quality assessment of error concealed depth maps.

### 1.2.1 Spatial and inter-view depth map error concealment

These contributions are classified by the following two groups of error concealment techniques: (i) spatial depth map error concealment, where spatial information of the correctly decoded neighbouring is used to recover the missing regions; (ii) inter-view depth map EC, where similarities between the corrupted region and the corresponding data from adjacent views are exploited. Specifically, the following scientific contributions were achieved.

The first developed method, is based on the recovery of depth contours within the lost regions. By extracting the contours and recovering their lost segments based on the Bézier curve fitting method, spatial interpolation is improved by using such contours to reconstruct the missing depth values. In this method, the contour smoothness characteristics are maintained and the contours are used as boundary limits of the homogeneous depth regions, which are then filled using weighted interpolation [10].

In the second method, the underlying objective is also focused on depth contour reconstruction. In this case, the similarities between texture images and corresponding depth maps is investigated. The method relies on the texture image to reconstruct the lost contour segments in the corresponding depth map areas. Such reconstructed depth map contours are then used as boundaries, at different depth planes, to recover the missing depth values, also using weighted interpolation [11].

In the third method, based on the combination of the two previously described ones was further investigated. In this case, information obtained from the texture image is used to perform a geometric curve fitting of the lost contour segments in the corresponding depth map areas [12, 13].

In the fourth method, a novel three-stage processing algorithm was developed using a combination of an inter-view depth reconstruction technique with the methods described in previous contributions [10, 11]. Initially sharp depth transitions are reconstructed using a disparity map, computed from the texture stereo pair. Then contour reconstruction of arbitrary shapes, within the lost regions of a depth map is combined with selective weighted interpolation in the spatial domain [14, 15].

In the fifth method, an inter-view-based technique is proposed using warping vectors obtained through a block matching approach using geometric transforms (BMGT) between

two texture views. It was found that BMGT, is capable of finding efficient warping vectors for reconstruction of lost regions in depth maps associated with texture views, even in complex camera arrangements [16].

The contributions in the topic of *Spatial and inter-view depth map error concealment*, resulted in the following publications:

**[10]** S. Marcelino, P. Assunção, S. Faria, and S. Soares, Error recovery of image-based depth maps using Bézier curve fitting, in International Conference on Image Processing (ICIP), September 2011.

**[11]** S. Marcelino, P. Assunção, S. Faria, and S. Soares, Lost block reconstruction in depth maps using color image contours, in Picture Coding Symposium (PCS), May 2012.

**[12]** S. Marcelino, P. Assunção, S. de Faria, and S. Soares, Efficient depth error concealment for 3D video over error-prone channels, in IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), June 2013.

**[13]** S. M. Marcelino, P. Assunção, S. Faria, and S. Soares, Two-stage depth map error concealment using geometric fitting, in Conf. on Telecommunications (ConfTele), May 2013.

**[14]** S. Marcelino, P. Assunção, S. Faria, and S. Soares, Spatial error concealment for intra-coded depth maps in multiview video-plus-depth, Submitted to Multimedia Tools and Applications, Springer, 2015.

**[15]** S. Marcelino, P. Assunção, S. Faria, L. Cruz, and S. Soares, Slice loss concealment for depth maps in multiview-video plus depth, in Conf. on Telecommunications (ConfTele), Setpember 2015.

**[16]** S. Marcelino, P. Assunção, S. Faria, and S. Soares, Depth map concealment using interview warping vectors from geometric transforms, in International Conference on Image Processing (ICIP), September 2013.

### 1.2.2 Temporal and MDC error concealment techniques for depth maps

The first contribution consists of a novel method to recover lost regions in depth maps of MVD. Recovery of lost depth regions is achieved by using geometric fitting based on inter-view/inter-frame methods, which compute a set of warping vectors obtained through a region matching approach with geometric transforms (RMGT), using the texture views and the depth maps of the same views. Additionally, the proposed method uses a contour reconstruction technique for interpolation of arbitrary shapes within lost regions. For packet loss rates (PLR) up to 20%, the proposed error concealment method is able to significantly improve the quality of the synthesised images. In comparison to the best reference method used in this work, the results show that an average PSNR gain of 1.48dB is obtained for the synthesised views of a video sequence [17].

The second contribution presents a method to improve the quality of depth maps transmitted as multiple descriptions (MDC) through multipath error prone networks. In MDC whenever a single description is lost, the remaining ones are still able to provide a coarsely decoded version of the depth map. While in multiple description video, such coarse decoding is still acceptable for display, in the case of depth maps, the additional decoding distortion propagates through the corresponding view synthesis with great impact in perceived quality. The proposed method improves low quality depth maps decoded from one single description, based on geometric information available in coarsely decoded slices, which are combined with higher quality depth values from adjacent slices, previously decoded using both descriptions. In comparison with existing MDC decoders, the enhancement method achieves quality gains for the synthesised views up to 1.69dB, for 10% packet loss rates [18].

As the previous contribution [18], the third contribution aims to efficiently reconstruct lost descriptions in MDC depth maps. In order improve depth maps reconstruction quality, the proposed method extracts geometric information from the received descriptions and motion information from texture. Thus, using information from depth map edges, spatially neighbouring depth values and texture motion, it is possible to recover lost or coarsely decoded depth maps from one single description. When compared with current MDC decoding, without enhanced reconstruction of lost descriptions, the proposed method presents objective quality gains up to 2.29dB, for packet loss rates of 10% [19].

The contributions in the topic of *Temporal and MDC error concealment techniques for depth maps*, resulted in the following publications:

[**17**] S. Marcelino, P. Assunção, S. Faria, and S. Soares, A method to recover lost depth data in multiview video-plus-depth communications, Submited to Journal of Visual Communication and Image Representation, ELSEVIER, 2015.

[**18**] P. Correia, S. Marcelino, P. Assunção, S. Faria, S. Soares, C. Pagliari, and E. da Silva, Enhancement method for multiple description decoding of depth maps subject to random loss, in 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), IEEE, July 2014.

[**19**] S. Marcelino, P. Assunção, S. Faria, and S. Soares, Depth maps reconstruction in MDC with lost descriptions, in Submitted to IEEE International BMSB 2016.

### 1.2.3 Quality assessment of error concealed depth maps for MVD

The contribution in this field refers to a quality evaluation study on the performance of error concealment methods for depth maps used in MVD [20]. This research deals with the problem of decoding corrupted depth maps received from error-prone networks, where the quality of the reconstructed depth data is not always directly related to the quality of the synthesised virtual views. Even after error concealment, depth map distortions are not individually perceived in the synthesised images as other known types of distortions, such as coding distortion. Thus, traditional quality metrics might not be adequate to capture all the relevant distortion features. In this work, the performance of two error concealment methods for depth maps is evaluated using a novel perceptually-aware objective metric (pPSNR). This metric is validated through subjective assessment of virtual views synthesised with recovered depth maps. Each subjective test was performed by comparing the relative quality between two synthesised images using different error concealment methods. The perceptual impact of using corrupted depth maps in MVD decoders with view synthesis is evaluated under various loss rates, using several texture images and depth maps encoded using multiple quantisation steps. The results reveal that the proposed objective quality metric is mostly in line with user preferences, in respect to the relative performance of each error concealment method.

The contributions in the topic of *Temporal and MDC error concealment techniques for*

*depth maps*, resulted in the following publication:

[**20**] S. Marcelino, S. Faria, R. Pepion, P. L. Callet, P. Assunção, and S. Soares, Quality evaluation of depth map error concelment using a perceptually-aware objective metric, in 3DTV-Conference: Immersive and Interective 3D Media Experience over Networks (3DTV-CON), IEEE, July 2015.

## 1.3 Thesis structure

This thesis is comprised of seven chapters presenting a research study on error concealment and restoration techniques for decoding corrupted depth maps using the MVD format. Throughout this thesis several original contributions, performance evaluation results and critical discussion are presented. In Chapter 1 the context and motivation are outlined, followed by a brief overview on the original contributions of the author.

In Chapter 2, an overview of the current 3D/Multiview formats and coding methods is presented. Firstly, the most common 3D video representation formats are described, such as Multiview Video (MVV), Multiview Video-plus-Depth (MVD) and Layered Depth Video (LDV). Secondly, the standard coding schemes for stereo and multiview video are presented, such as Simulcast, Stereo Interleaving, Multiview Video Coding (MVC) and Multiview Video plus Depth Coding (MVC+D).

In Chapter 3, a review of the current state-of-the-art in the field of error concealment techniques is presented. This chapter is organised in four sections, each one corresponding to a different category of error concealment techniques. The first category addresses basic methods presented in the literature for error concealment of 2D video in the spatial and frequency domains. Temporal error concealment techniques are presented as part of the second category, namely Frame Copy (FC), Motion Copy (MC), Motion Boundary Matching Algorithms (MBMA) and Motion Vector Extrapolation (MVE). The third category focus error concealment techniques for stereo and multiview video and finally, recent error concealment techniques specifically tailored to depth maps are presented, with emphasis on those more relevant for this thesis.

Chapter 4 presents the novel methods investigated and developed in the scope of this thesis. As mentioned before, these methods are based on spatial domain and inter-view error

concealment techniques. This chapter is organised in five sections, each one presenting an error concealment technique specifically investigated for depth maps in MVD. Three of them are based on the reconstruction of corrupted depth map contours. The first technique uses curve fitting with Bézier curves to recover the missing contour segments; the second method uses the texture image contours associated with the corrupted depth map to reconstruct lost contours; the third technique is a combination of the previous two. In the fourth section, inter-view similarities are exploited using disparity information to recover lost depth maps. Finally, inter-view similarities are also exploited in the method described in the fifth section, using a block matching algorithm based on geometric transforms (BMGT).

In Chapter 5, the new temporal error concealment techniques developed for depth maps in MVD and Multiple Description video Coding (MDC) are presented. This chapter is organised in three sections, also corresponding to three different techniques. In the first section, a slice-based temporal error concealment technique is proposed, based on BMGT. Both inter-view and inter-frame similarities between depth maps and the texture images are exploited. In the second section, an error concealment technique is used for depth maps encoded with MDC, when a single description is lost. This technique relies on a spatial interpolation technique, based on the geometric characteristics of the depth maps. In the third section, a method to further enhance MDC depth maps is devised by exploiting both spatial and temporal similarities, which allow concealment of large missing regions, up to an entire depth map.

Chapter 6 presents a novel perceptually-aware objective quality metric for synthesised images, obtained from recovered depth maps. This chapter is structured into three sections: the first one briefly describes the error concealment methods applied to the depth maps used in this investigation; the second section presents the proposed objective quality metric; the third section describes the experimental setup used to evaluate the accuracy of the proposed metric with subjective tests, results and discussion.

Finally, in Chapter 7 a discussion about the accomplishments of this dissertation is provided along with identification of open topics for future research.

# 2 3D/Multiview formats and coding methods

This chapter, describes the most common 3D displays, 3D content representation formats and the corresponding state-of-the-art coding methods. Stereoscopic video and multiview video are presented and then extension of these formats to multiview video-plus-depth and layered depth video is introduced. Regarding coding methods, a technical overview is provided, focusing simulcast systems and their evolution to more efficient ones, such as stereo and multiview video codecs, multiview-plus-depth coding and finally specific methods used for depth coding.

## 2.1 3D displays

3D displays can be classified in two main categories: classic stereoscopy with two views and displays with parallax property, where the user has the ability of viewing different perspectives throughout the scene. In this overview, a brief description of some of the most common devices of each type is given [21, 22].

### 2.1.1 Stereoscopic displays

Regarding stereoscopic displays, this type of technology relies on the basic stereoscopic principle that one different image is delivered to each eye. Those devices can also be

sub-divided into two other groups: 3D with either passive or active glasses and auto-stereoscopic displays.

• **Displays with anaglyph systems.**

One of the most classic and well known forms of watching stereoscopic content is based on anaglyphes. This technology relies on two images that are encoded in different colour channels, in order to include two separate images superimposed in the same picture which are necessary for stereoscopy. The visualization is performed by using a pair of glasses similar to the ones shown in Figure 2.1, which contains a different filter in each lens, corresponding to the colour channels that were used to encode the stereo pair. When the user is wearing these glasses and watching a 3D picture/movie in this format as shown in Figure 2.2, he/she is able to experience stereopsis and to have depth perception. This approach was very popular in the past due to its cheap construction and implementation, but in the last few years, with the emergence of new technologies that are now available and more affordable do general public, the classic anaglyph method is becoming obsolete. The main disadvantages of this approach are result from the colour filter separation technique. Such technology is not able to to deliver full 3D colour images to the viewer and high crosstalk levels are present, which is known to be one of the most uncomfortable distortions in 3D [23].

**Figure 2.1** – Anaglyph glasses.

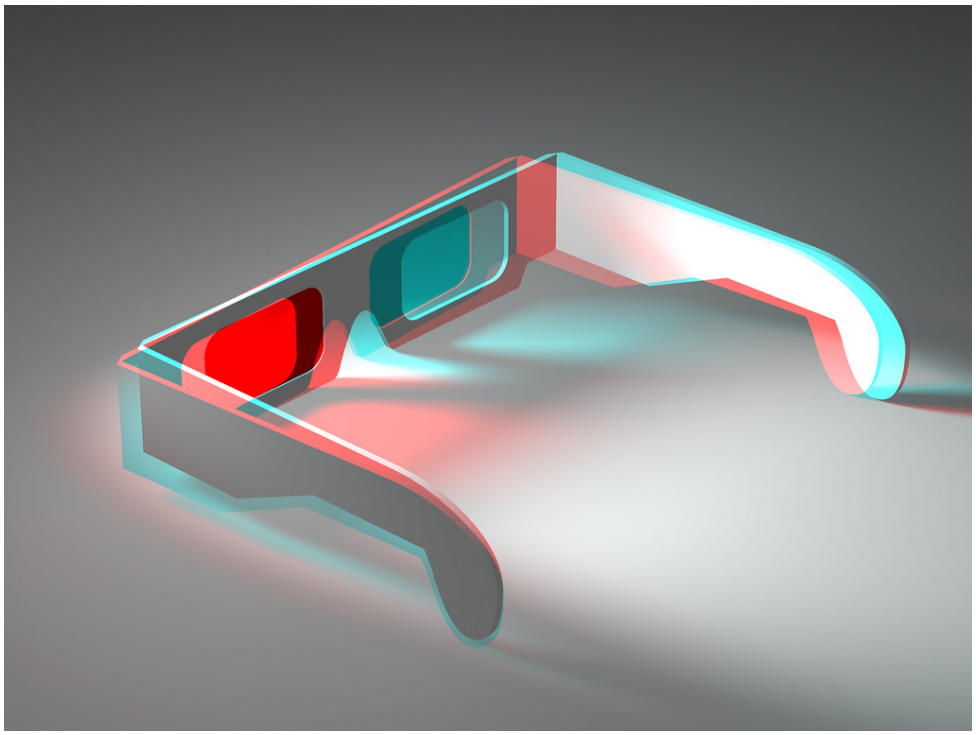


**Figure 2.2** – Example of an anaglyph image.

- **Displays with polarized glasses.**

Stereoscopy using polarisation filters is a very popular solution in 3D cinema. In the last few years it is becoming also very popular in home TV and cinema, but it is still a relatively expensive technology, which limits its popularity among the general public. This type of technology relies on the principle that a pair of images that are projected through polarizing filters, which are in conformity with the glasses of user, leads to a separate image displayed to each eye simultaneously. There are two main polarizing techniques, linear-polarized displays and circular polarized displays [24].

In the linear polarized displays, each left and right images are projected through orthogonal filters at different angles for each eye and the passive glasses have filters to match to those orthogonal filters, enabling stereoscopy. Since the polarizing filter has a specific angle for each eye, the user cannot move his head significantly because this can cause a substantial tendency to appear crosstalk resulting in ghosting artefacts.

Circular polarized displays are much more effective and much less sensitive to the user's head position due to the characteristics of circular polarization. In this case, both views are simultaneously projected through a pair of circular filters. Figure 2.3 shows an example of stereoscopic display with a retarder film (PRF). Odd lines correspond to the right circular polarization, while even lines correspond to left circular polarization. The main disadvantages of the polarizing approaches, comparing to the anaglyph, is the need for projecting the two views at the same time, which leads to half of the resolution for the 3D image. The advantage of circular polarization compared to linear polarization is reducing significantly the artefacts that were mentioned in the previous display polarization approach.

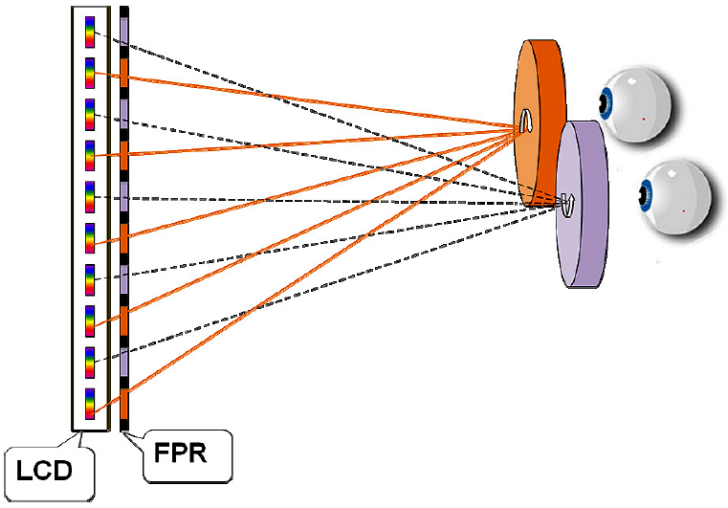



**Figure 2.3** – Circular polarization example [24].

One of the main advantages of polarized displays is the cost of the glasses which is very affordable compared to the active ones. On the other side, the display is much more expensive when compared to a conventional 2D display. This is due to the polarized filter techniques that are used with the spatial multiplexing of the views.

- **Displays with active glasses.**

To achieve stereoscopy, displays with active glasses relies on temporal multiplexing of both left and right views. The active glasses, also known as *shutter glasses*, are synchronized

with a conventional 2D display, blocking each eye in every other image, so that the user can alternatively receive the correct image of the stereo pair in each eye. The switching of the active glasses, synchronized with the display, must be fast enough so that the user cannot perceive the opening and closing of the active glasses, resulting in the perception that the viewers are receiving different images to each eye simultaneously. Besides the synchronization device that integrates the glasses with the conventional 2D display, the display must have the capacity to output a higher frame rate that can cope with the glasses switching rate that should not be perceived by the user, e.g. 120Hz.

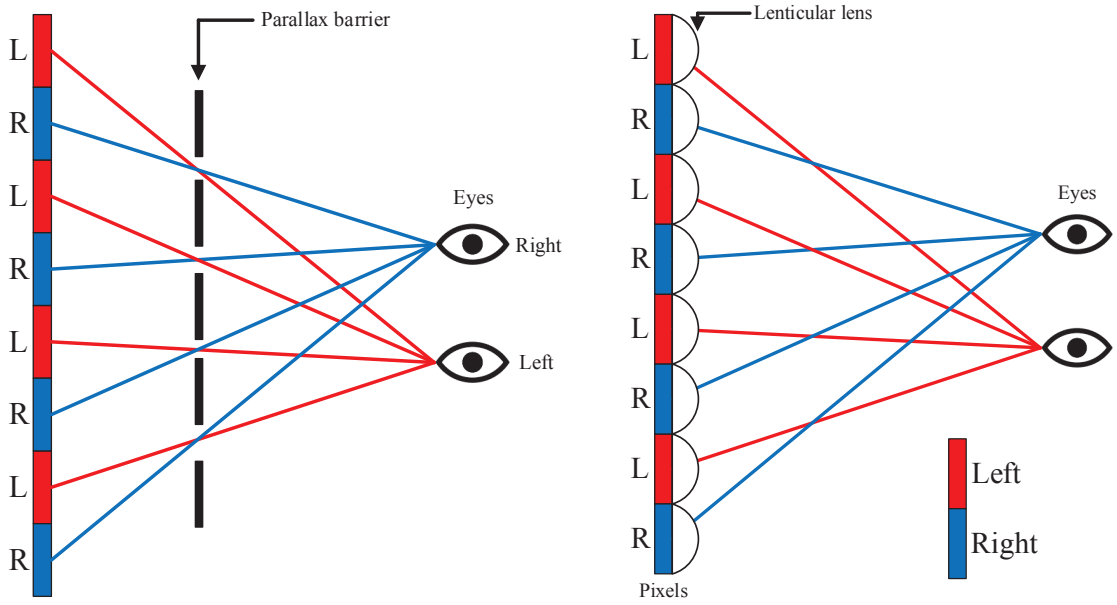
The main disadvantages of this type of systems consists in the higher complexity of the active glasses, in comparison with passive glasses, which results in heavier devices, also requiring regular recharging. However, depending on the lighting conditions, the user may experience some flicker artefacts. The main advantages are the improved 3D experience, even when the user moves his/her head and the possibility to use existing 2D displays with an external syncing device, where the only requirement is the higher display frame rate that must be compatible with the glasses switching rate. Having in mind that today, most conventional home screens, have a response time significantly lower than 10ms, they are able to easily deliver twice the frame rate of normal 2D video.

- **Auto-stereoscopic Displays.**

Auto-stereoscopic displays allow watching 3D content without any accessories such as glasses [25, 26]. In the last few years, this type of technology is growing significantly, mainly due to its lower cost which is much more affordable nowadays. The main advantage is the increased comfort that results from the fact that viewers do not need to wear any kind of accessories. The main disadvantages of auto-stereoscopic displays is the limited number of viewing positions, despite the increased mobility of the viewer because it does not have to wear glasses. In general, the image quality is also inferior than displays that use glasses because the effective resolution of auto-stereoscopic is lower than conventional ones. Current consumer auto-stereoscopic displays rely on two main technologies, based on parallax barriers and lenticular lens as shown in Figure 2.4.

Regarding the parallax barrier technology, Figure 2.4(a) shows an example of how a stereo pair is directed to the viewer's eyes. The left (L) and right (R) pixels are redirected by the barrier, as shown by the blue and red rectangles. Then a barrier (parallax barrier)

has evenly spaced openings in order to separate left from the right pixels. If the viewer is positioned in the right place in front of the screen, the left image will be aligned with the left eye and the right image will be aligned to the right one.




(a) Parallax barriers. (b) Lenticular lens.

**Figure 2.4** – Auto-stereoscopic displays.

In the second case, the principle of aligning the left and right pixels to the corresponding eyes is similar to the parallax approach, but the system used to perform that task is different. As show in Figure 2.4(b), a lens on each pair of L and R pixels is used to focus the correct images to the corresponding eyes. This technology allows having multiple viewpoints more effectively than the parallax barrier technique and also allows a certain degree of horizontal parallax. This characteristic is also very appealing for FTV applications where horizontal parallax is one of the main features [4].

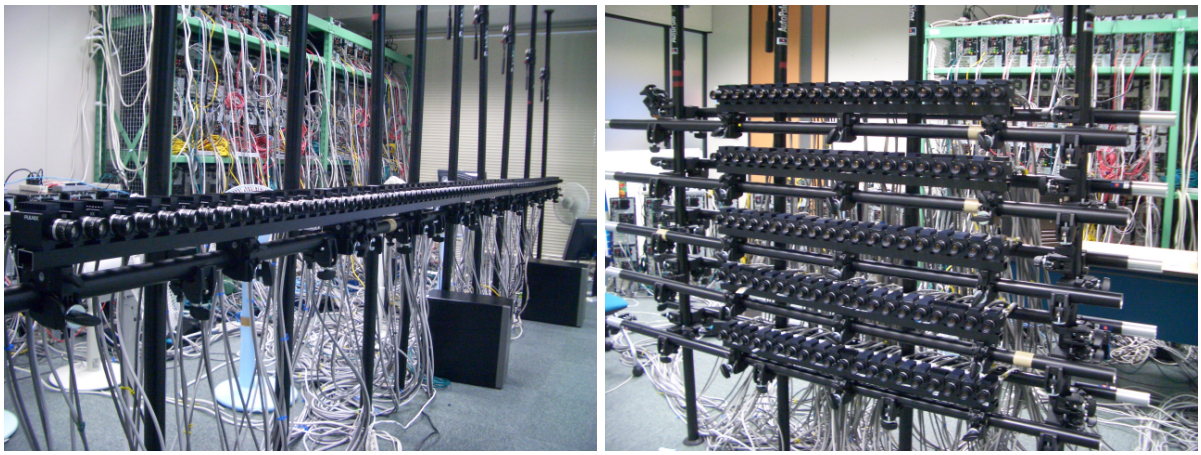
## 2.2 3D formats

### 2.2.1 Multiview Video

To extend the range of 3D applications beyond simple stereoscopic image/video, it is necessary to use more than two views in order to increase the number of viewpoints

available for a 3D scene. For example, applications such as Freeview Television (FTV) [4] on autostereoscopic displays require a large number of views to actually provide an immersion feeling to users. These emerging types of 3D applications have the ability to provide advanced features to the users, such as motion and stereo parallax. The downside is the huge amount of data that needs to be captured, processed and transmitted, due to the large number of views that is required to provide a good user experience.

In the Multiview Video format (MVV), each view is captured by a different acquisition camera. A set of cameras is normally organized in geometric structure, where the most common arrangements are the linear arrangement, circular arrangement and 2D arrangement, as shown in Figure 2.5. These camera systems were developed in Fuji Laboratory at Nagoya University [27], where some MPEG test sequences were acquired. Some of these sequences were used in the work described in this thesis.



(a) Linear. (b) 2D.

(c) Circular.

**Figure 2.5** – Example of different camera arrangements [27].

Since in the MVV format each view delivered to end users is captured from a real camera (not virtual), the amount of data that needs to be transmitted or stored is very high. It can be easily shown that the amount of data needed to represent an MVV scene increases linearly with the number of views [28]. Thus, if the use of such format generalizes, the total number of views that can be delivered is necessarily limited by bandwidth constraints or available storage capacity.

### 2.2.2 Multiview Video-plus-Depth

In recent years, the display capabilities have improved significantly and the current technology is capable of exhibiting a higher number of views. Due to the inefficiency of MVV to represent a dense number of views, multiview video-plus-depth (MVD) emerged to overcome this problem by enabling the synthesis of virtual views from the ones actually captured from the scene. In MVD, each texture image is associated with a corresponding depth map, which is represented as a greyscale image of the same resolution, where each sample of the depth map is the distance between the co-located pixel in the camera and its point in the scene, as illustrated in Figure 2.6. In the presence of a texture image and such geometric information represented by the depth map, it is possible to synthesise additional virtual views. Thus, using MVD, significantly allows reducing the amount of data required to represent multiple views, when compared to MVV. Because the depth map is represented by a smooth greyscale image, the compression efficiency is much higher than the corresponding texture image, while several different virtual views can be generated [29].

In video delivery services, since the virtual views are synthesised at the decoder side, the MVD format has an important advantage over MVV, that is the user experience may be improved by increasing the number of virtual views without requiring extra bandwidth for transmission of extra views. This type of flexibility provides richer 3D content than in the case of MVV, where the only views available for display are those captured, encoded and transmitted to the user equipment.

A depth map is usually represented as an 8-bit greyscale image, corresponding to a depth range from 0 (far from the camera) to 255 (near the camera). Depth information can be either acquired by depth sensors or can be computed from disparity information [31].

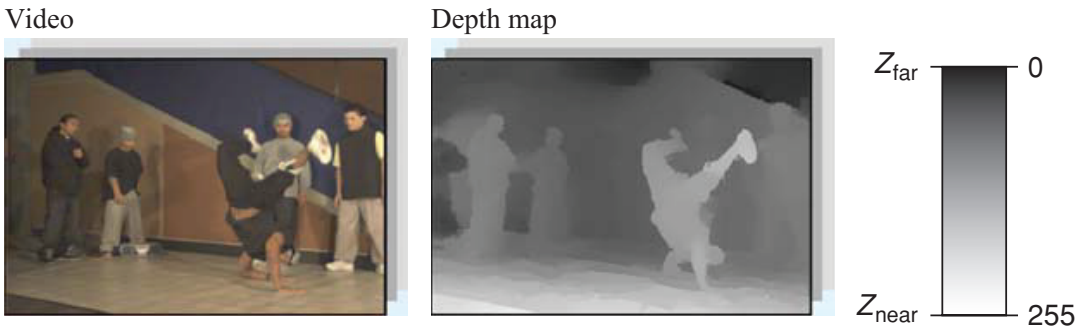



**Figure 2.6** – MVD: Video image and corresponding depth map (Breakdancers sequence) [30].

Depth maps obtained from sensors, normally use *time-of-flight* technology [32], which is a low complexity task, allowing to acquire depth information in real time. This type of sensors are nowadays quite common, especially in home entertainment applications, as they exhibit good performance in indoor/confined spaces and also a reasonable cost. However, to produce 3D content with professional quality, these types of sensors have major limitations because the accuracy of depth data is not very high due to significant noise and interferance from other light sources, e.g. sunlight. Despite being filtered using post-processing techniques, usually loose depth details are lost using this type of technology. *Time-of-flight* sensors also present a quite low resolution, such as 240×240, pixels [33], which is also a limitation because accurate upsampling processes are required to reach the same resolution as the corresponding images.

Other methods used to obtain depth maps compute depth through disparity matching techniques based on two texture views. For example, the authors in [31] use the widely known relation between depth maps and disparity maps, given by Equation 2.1, to compute depth from disparity obtained by stereo matching algorithms.

$$Z(u,v) = \frac{r.F.B}{D(u,v) + D_O(M_C)}, \tag{2.1}$$

In the above equation, the depth map is represented by $Z(u,v)$, and the disparity map by $D(u,v)$. $r$ defines a scaling factor caused by de-rectification, $D_O$ defines de the disparity offset from a given convergence point $M_C$. Finally, $F$ and $B$ are the baseline and focal length, respectively, of the rectified stereo pair being considered.

### 2.2.3 Layered Depth Video

Layered Depth Video (LDV) can be interpreted as an extension of the Layered Depth Image (LDI) format [34], which was previously presented for stereo [35] and multiview [36] applications. As mentioned before, MVD is a very practical and effective video format for 3DTV and FTV, but when used in application scenarios where multiple views are generated from distant viewpoints the occluded regions of the virtual views tend to be significant.

In order to overcome this problem, in LDV, the residual data corresponds to the occluded areas between views [34], as shown in the LDV images of Figure 2.7. The accuracy of virtual view synthesis with traditional MVD has the disadvantage that the occluded regions are filled using uncertain estimates, which typically introduce visible artefacts. The occluded regions in LDV are available in additional layers to enhance the synthesis accuracy.

Figure 2.7(a) and 2.7(b) show the corresponding warped (i.e., virtual view without occluded regions) texture and depth maps. The corresponding residual data texture and depth are, respectively, shown in Figures 2.7(c) and 2.7(d). The blank regions in the warped texture image (i.e. occluded areas) and depth map can be filled with corresponding residual data.

(a) Central warped texture from left and right view. (b) Central warped depth from left and right view.

(c) Residual texture data. (d) Residual depth data.

**Figure 2.7** – Example of LDV images and respective residue [34].

## 2.3 3D/Multiview Coding methods

As mentioned in the previous section, there are several encoding methods and standards to encode 3D visual content according to its representation format. In classic stereoscopic video only two views are required, but for more complex formats, as used in multiview auto-stereoscopic displays and FTV display systems, a larger number of views is required. Thus, different encoding methods are used in order to match the specific characteristics of each format and to obtain increased efficiency.

For stereoscopic video, simulcast and stereo interleaving are common methods used to encode and deliver 3D video. When a larger number of views is used, more sophisticated coding/delivery methods must be used in order to reduce the amount of data to transmit and/or store. In these cases, Multiview Video Coding (MVC) and Multiview Video Coding plus depth (MVC+D) are used, in order to exploit the redundancies between views and depth maps, respectively for MVV and MVD formats. In this sections, a brief description of the most important 3D video coding schemes is presented, as part of the study carried out in this thesis.

### 2.3.1 Simulcast

Simulcast is a simple method used for transmitting video with multiple views, using existing standards, primarily designed for 2D video. In simulcast, each view is independently encoded and transmitted, as shown as the coding structure in Figure 2.8. Since no correlation between views is exploited to maximize the coding efficiency, the amount of transmitted data is $N$ times ($N$ = number of views) the size of one single 2D video stream. The advantages of using such a coding scheme, besides its simplicity, is the error resilience and free switching between views in interactivity services. Since views are coded independently from each other, transmission losses in one view do not affect the others, which might be useful to guarantee at least 2D video at the receiving side. From the point of view of user interactivity, independent views allow users to switch between any views, without inter-view delays, because they respective streams are separately decoded.

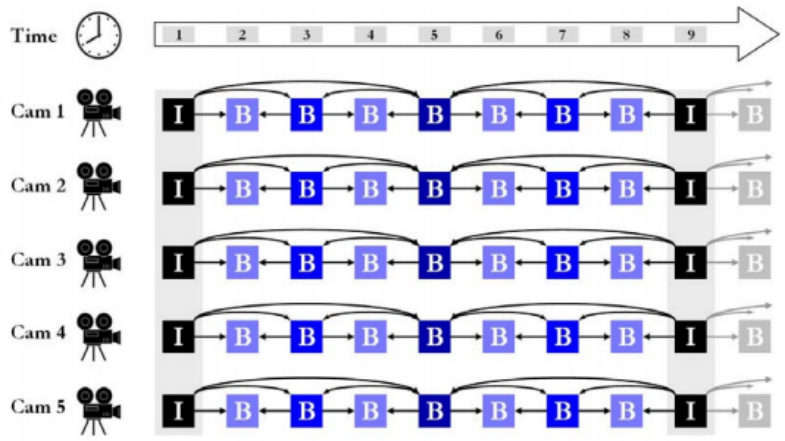



**Figure 2.8** – Example of a simulcast coding scheme [37].

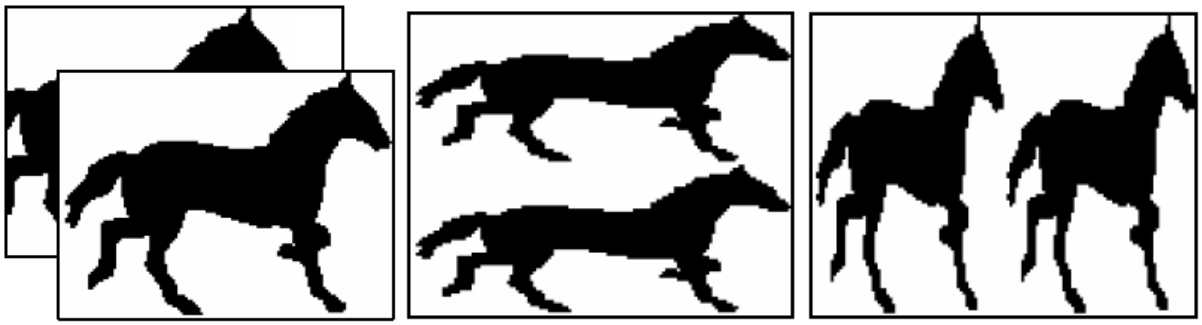
Another advantage of using simulcast is the non-existence of coding dependencies between views, which allows easier exploitation of infrequent data networking scenarios, such as multi-path and heterogeneous networks. To minimise the problem of reduced coding efficiency in simulcast, several techniques based on asymmetric 3D video coding might be used. Asymmetric 3D video coding relies on characteristics of the Human Visual System (HVS) [38] that, for a given stereoscopic content with different quality for each view, the perceived quality will be dictated by the view with higher quality [39]. Asymmetric coding can be classified into two main categories: Mixed Resolution and Asymmetric Quality. In the first category, both spatial and temporal resolution can be different between each view

[40, 41]. In the latter, different quality can be used in each view by assigning different quantisation step (QP) or by allocating different bit-rates [42, 43].

### 2.3.2 Stereo interleaving

Stereo interleaving is an effective method to take advantage of the systems, originally designed for 2D video [44]. There are two main types of stereo interleaving modes: Temporal and Spatial, as shown in Figure 2.9. In temporal interleaving (Figure 2.9(a) ), each left and right view image is encoded and transmitted alternatively along time. In spatial interleaving, the most common methods encode two views simultaneously on the same image using side-by-side (Figure 2.9(b)) or top-and-bottom (Figure 2.9(c)) structures.

It is relatively easy to adapt the current 2D systems to support this form of stereo media delivery. When a stereo video signal is transmitted, the receiver has to be signaled about the presence of 3D content and, thus, it must receive information about the format that is being used in the coded stream. In the current video coding standards, additional signaling information for stereo video is transmitted through Supplementary Enhancement Information messages (SEI).



(a) Temporal interleaving. (b) Spatial interleaving, top-and-bottom. (c) Spatial interleaving, side-by-side.

**Figure 2.9** – Three examples of stereo interleaving [45, 46].

In the case of temporal interleaving, the transmission channels have to support higher bit rates than necessary for monoscopic video at the same frame rate. Otherwise, there is a temporal sub-sampling. In spatial interleaving, images have to be spatially sub-sampled in order to be compliant with the scheme shown in Figure 2.9 b, c). The main drawback of this solution is the reduced spatial resolution that results from the sub-sampling process,

which leads to a lower quality when compared with full-resolution images.

### 2.3.3 Multiview Video Coding

Multiview Video Coding (MVC) is an extension of the H.264/AVC standard and more recently, also of MV-HEVC was developed as a multiview extension of H.265/HEVC. These extensions were specifically tailored for encoding multiple views with higher efficiency than simulcast [7, 47]. The main features of this extension are the backward compatibility with H.264/AVC (or H.265/HEVC) and the inter-view prediction, to exploit redundancy between views. As mentioned before, each view is acquired by a camera that captures different views of the same scene, so there is inherent redundant data in the digital representation of adjacent views. MVC and MV-HEVC take advantage of this characteristic by exploiting inter-view prediction besides temporal redundancy. In order to increase the coding efficiency of MVC/MV-HEVC, it is of major importance to choose efficient prediction structures. Relevant results can be found in [48].

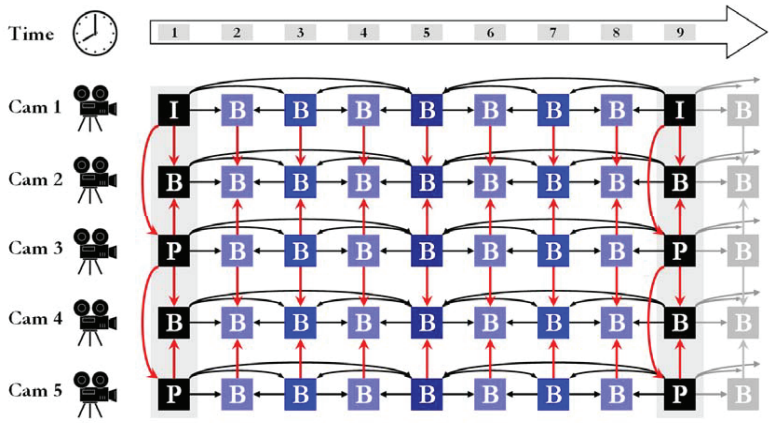



**Figure 2.10** – Multiview Video Coding (MVC) structure [37].

Figure 2.10 shows an example of a MVC coding structure, with a setup of five views (Cam 1 to Cam 5) and a Group Of Pictures (GOP) structure of eight frames. Note that the MVC extension introduces a new type of picture: the anchor pictures. This type of pictures has some characteristics in common with I/IDR pictures of H.264/AVC, because they do not allow temporal predictions, only inter-view predictions in order to support backward compatibility with H.264/AVC. The first view in MVC is encoded independently from the others and it is named as the base-view. The other views are encoded sequentially,

according to the predefined coding structure, as exemplified in Figure 2.10, allowing the encoder to exploit temporal and inter-view redundancies. Note that the MVC and MV-HEVC specification does not allow inter-view prediction from different temporal instants. Additionally, all reference pictures for inter-view prediction must be included in the same access unit that corresponds to the NAL (Network Abstraction layer) of that specific instant.

The coding efficiency obtained with MVC, when compared to H.264/AVC simulcast, it is significantly superior [7]. For some cases, the quality gains over simulcast can be as high as 3dB, corresponding to a bit-rate decrease of roughly 50%. For a typical multiview video with eight views, the average bit-rate savings over simulcast is approximately 20%.

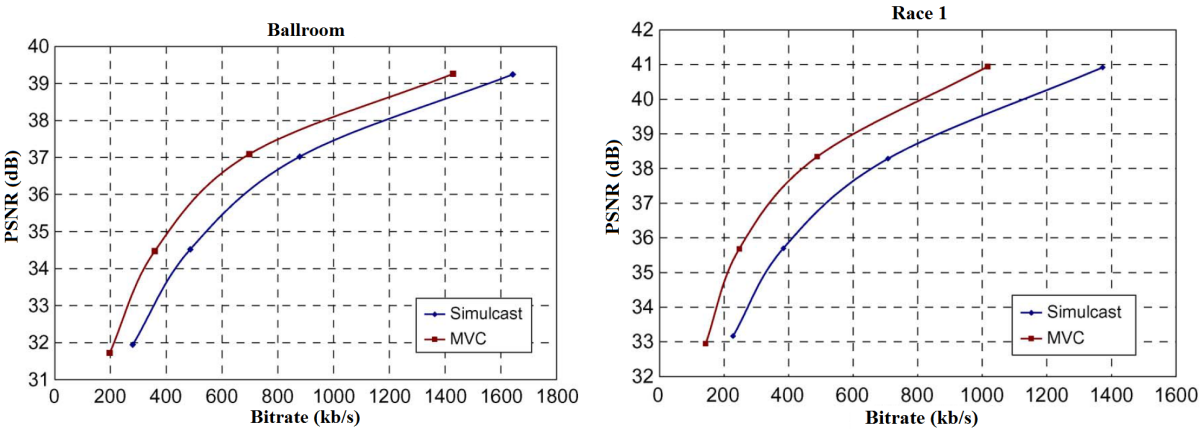



**Figure 2.11** – Rate-distortion example for several test sequences encoded with MVC (extension of the H.264/AVC): Ballroom and RaceFigure 2.11 [7].

Figure 2.11, shows the rate-distortion (RD) performance of two test sequences: Ballroom and Race1 [7]. These tests were obtained in conformity with the *common test conditions*, defined by the Join Video Team (JVT) group [49]. The encoding tests of these two sequences were performed by using both temporal and inter-view predictions, shown in the MVC coding structure of Figure 2.10. As described in the MPEG document [50], most of the gains obtained by MVC over simulcast result from the use of inter-view prediction in anchor pictures. If inter-view prediction is also used for non-anchor, it is expected to obtain an average improvement of 5-15% in bit-rate for the same quality, at a cost of a higher encoder/decoder complexity.

The standardised MVC/MV-HEVC offers a substantial degree of flexibility, being possible to adapt the coding structure in order suit the needs in terms of complexity and quality requirements. Since each view is encoded sequentially, this also allows to configure different coding parameters for each view, resulting in the ability to adopt asymmetric coding schemes [40, 51–55].

Regarding the most recent MV-HEVC, simulation results also shown an significant improvement in performance when compared with HEVC simulcast. The results presented in Table 2.1 show that an average bit-rate saving of 28% (two view) and 38% (three view) when comparing with HEVC simulcast.

**Table 2.1** – MV-HEVC VS. SIMULCAST: Bit-rate reduction [47].

| Sequence | View 1 only | View 2 only | Total 2-view | Total 3-view |
|---|---|---|---|---|
| Balloons | 53.9% | 49.7% | 23.5% | 31.5% |
| Kendo | 52.5% | 47.2% | 23.3% | 30.4% |
| Newspaper | 56.4% | 54.4% | 23.3% | 33.2% |
| GT_Fly | 82.0% | 81.3% | 38.7% | 52.4% |
| Poznan_Hall2 | 53.5% | 53.9% | 23.3% | 32.8% |
| Poznan_Street | 69.7% | 69.4% | 29.7% | 41.4% |
| Undo_Dancer | 74.5% | 76.0% | 34.0% | 47.3% |
| 1024×768 | 54.2% | 50.4% | 23.4% | 31.7% |
| 1920×1088 | 69.9% | 70.2% | 31.4% | 43.5% |
| Average | 63.2% | 61.7% | 28.0% | 38.4% |

### 2.3.4 Multiview Video-plus-Depth coding

As mentioned in Section 2.2.2, MVD is an efficient format that allows to synthesise virtual views at the receiver, hence reducing the number of views to be encoded and transmitted. Despite that depth data is not intended to be viewed by the user, its integrity is very important for the quality of the synthesised views. For this reason it is of utmost importance to use encoders that preserve the geometrical information of the scene given by the depth data [56].

Figure 2.12 shows several extensions of the existing coding standards that support multiple texture views and depth maps. MVC plus depth (MVC+D) is the extension of MVC that supports depth maps. 3D-AVC is the extension of H.264/AVC where depth maps

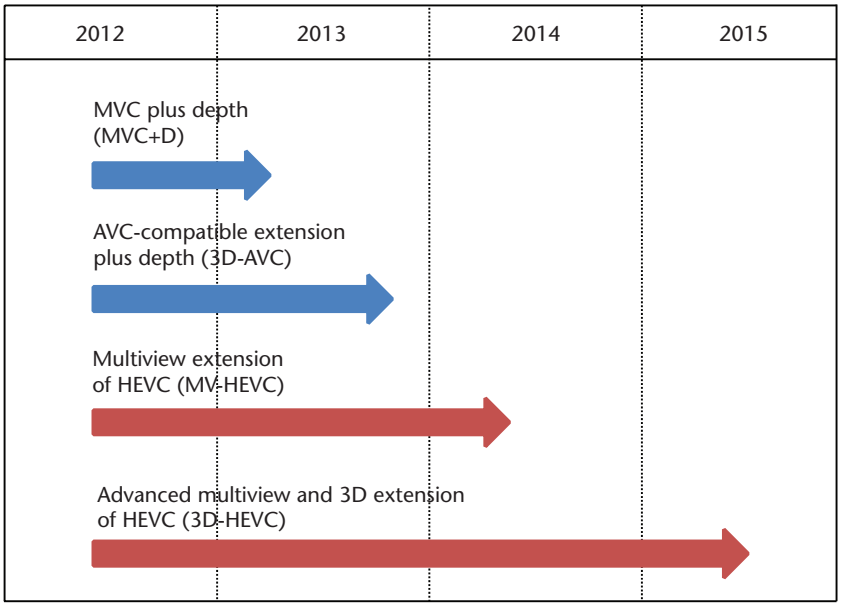



**Figure 2.12** – Extensions of the current coding standards (H.264/AVC and H.265/HEVC)for MVD [56].

are encoded in simlar manner as MVC+D but textures images are still compatible with H.264/AVC. 3D-HEVC is the extension multiview video with depth map support that derives from MV-HEVC standard [56–58].

### Texture coding

As mentioned before, for texture coding in MVD, besides temporal redundancy typically used in 2D video, inter-view redundancy is also exploited [56]. There are also few techniques that use depth data to enhance texture coding in order to improve the coding efficiency. Using depth information, the redundancy between depth and texture is either exploited or, by synthesising additional viewpoints, in order to create alternative references for both inter-view and inter-frame predictions. The approach of generating additional viewpoints in order to create new references is also widely known as View Synthesis Optimization (VSO). In [59], besides using disparity compensated prediction, the authors also use View Synthesis Optimization (VSO). It was demonstrated that the combination of these techniques results in an improved coding efficiency.

Depth maps can also be useful for texture coding, not only to improve coding efficiency

but also to reduce computational complexity. For instance, in [60], the authors presented a method to reduce inter-view prediction complexity. By using the geometry parameters of the multiview setup together with depth maps of the texture image being encoded, initial estimates of disparity vectors (DV), close to the optimal ones, can be obtained. This allows to reduce significantly the number of search points, resulting in a speed-up increase of about 4.2 times, when compared with the reference search techniques.

If texture and depth components are coded independently, then the compatibility with classic decoding methods is more straightforward [61]. However, the usage of new tools that were conceived for MVD results in a significant improvement in texture coding efficiency.

### Deph map Coding (DC)

In general, depth maps are composed of large homogenous areas with smooth variations, surrounded by sharp edges corresponding to the objects boundaries contained in the scene at different distances from the video camera. The encoding algorithms should take these characteristics into account to preserve geometric features of depth map, since it is of major importance to preserve the contours/edges that define object shapes. The shape boundaries correspond to high frequency content, and traditional transform-based encoding tends to blur these regions. The integrity of such sharps edges that define the object boundaries are critical to synthesise images with good quality and to avoid blending texture from background in foreground, and vice-versa. As depth maps are not to be displayed but used for view synthesis, instead, depth map quality should not be evaluated by itself but using the corresponding synthesised images [62].

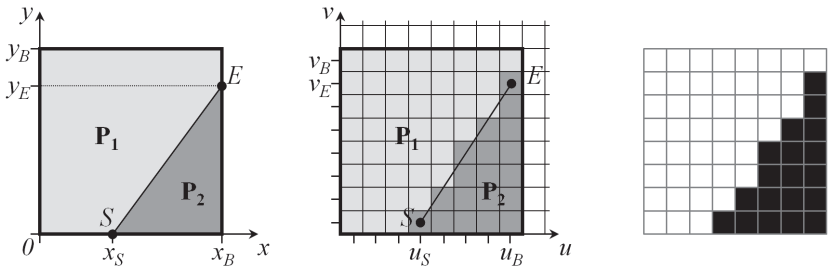
As mentioned before, texture and depth components may be encoded independently, or the information from one component may be used to improve the coding efficiency of the other one. In the 3D-AVC extension [63] the texture component is compatible with MVC [6], while depth map is encoded as in MVC+D [64]. In 3D-HEVC [47, 65], new features were adopted for depth map coding [56], which include depth motion prediction, Partition-based Depth Intra Coding and Segment-Wise DC Coding (SDC).

**Depth Motion Prediction:** Motion prediction in depth maps is performed in a similar manner as in texture coding. The main difference is the new set of motion vector (MV) candidates that are included in the candidate list of the motion estimation process. The candidate list is composed of a set of MVs that are used to define the initial search points, the better are these candidates, the faster is the motion estimation. In depth map coding, additional candidates are used: inter-view merge candidate, a sublock motion parameter inheritance candidate and a disparity-derived depth candidate [56].

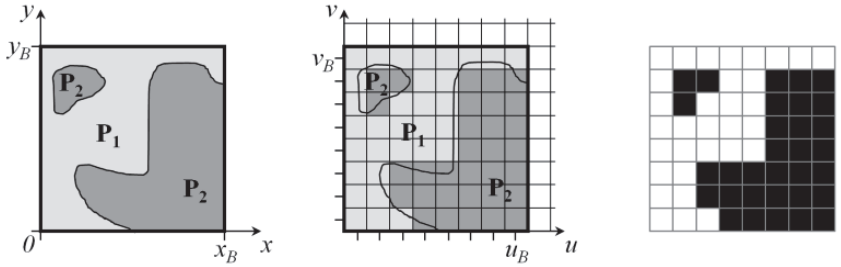
**Depth Look up Tables (DLT):** Despite the fact that depth maps are usually represented by a matrix of integer values within a range of [0-255], some of the recommended *JCT-3V 3DV* test sequences do not use the full range [0-255] [66–68]. This is because depth maps are usually highly quantised, resulting in fewer number of depth levels. To benefit from such characteristic, a Dynamic Lookup Table (DLT) can be computed in the encoder side by processing the set of depth maps contained in a group of pictures (GOP). The advantage of using DLTs is the reduced number of bits required to encode the residual values. The DLT is then carried on the picture parameter set (PPS), in both streams, i.e., the base and independent views.

**Partition-Based Depth Intra Coding:** To better represent specific characteristics of depth data, each depth block may be geometrically partitioned using intra partition modes for increased coding efficiency. In 3D-HEVC, these non rectangular partitions are referred to as depth modelling modes (DMM). In DMM it is considered that a certain block being encoded can be divided into two regions, defined by the depth map edges [69]. This technique allows the use of two types of partitioning patterns, as shown in Figure 2.13. The first is the Wedgelet pattern (Figure 2.13(a)), where segmentation of the depth block is defined by a straight line. The second is a contour pattern (Figure 2.13(b)), where the block segmentation relies on the block contours, allowing irregular partitions.

In the example of Figure 2.13 a given region is partitioned in two sub-regions, resulting in partition $P1$ and $P2$ (Wedgelet). On the left side of Figure 2.13(a), such separation can be easily represented by a linear equation using a line defined by points $S$ to $E$. In a practical scenario, a discrete signal is used as shown in the middle of Figure 2.13(a). To further use this information in the encoding process, the partitions are stored as binary maps with size of $u_B \times v_B$ as shown on the right side of Figure 2.13(a).




(a) Wedgelet block partition, continuous (left), discrete signal (middle) and corresponding pattern (right)




(b) contour block partition, continuous (left), discrete signal (middle) and corresponding pattern (right)

**Figure 2.13** – Partition-Based Depth Intra Coding and the corresponding partition pattern [69].

In the case shown in Figure 2.13(b), block partitions are defined based on the depth contours. Thus, the partition is more complex than in Wedgelet because the separation cannot be defined based on simple linear equations. This extra difficulty arises from the fact that the depth contours are arbitrarily shaped and can even result from a composition of several parts. Besides this fact, the contour partitions and Wedgelet are very similar and there is a clear advantage of the contour technique in regard to its ability of representing more complex shapes because it represents more efficiently arbitrary shapes. The values $P1$ and $P2$ are then computed based on the DLT technique, that was previously described.

**Segment-wise Depth Coding (SDC):** This coding mode enables to skip the transform and quantization process, such that the depth prediction residuals are directly coded. It also supports a depth look-up table (DLT) to convert depth map values into a reduced dynamic range. SDC can be applied to both intra and inter prediction, including DMM modes. When the SDC mode is applied, only one DC-coefficent predictor is computed for

each partition, and thus only one value is encoded to represent the residue of the whole partition [67, 70].

**Non-standard depth coding techniques:** Besides the work emerging from the standardization groups, other research efforts have been made in order to develop efficient coding algorithms to deal with the specificity of depth maps. This has been mainly achieved by introducing new intra prediction techniques, which allow to improve the coding efficiency of depth maps and the quality of the synthesised views.

In [71] a new intra coding technique was proposed, where each MB is divided into two flat regions using the depth map edges. This allows to represent an edged MB with arbitrary shapes more efficiently than the rigid partitions used in H.264/AVC. Note that this new intra mode was added to the existing H.264/AVC intra modes, and the best mode is chosen according to a Rate-Distortion (RD) optimization criterion. Edge information is encoded using context coding with an adaptive template. The edge structure of previously decoded MBs is also taken under consideration in order to further increase coding efficiency. The authors report improvements in the quality of the synthesised views and savings up to 25% in Bjöntegaard [72] bit-rate (BDBR) .

In [73, 74] an edge-aware intra prediction mode for DC coding is presented. This method applies intra directional predictions, based on the H.264/AVC intra mode directions, to the extracted edges of the depth map. Only the intra directions producing low residual energy are selected. Typically, a H.264/AVC intra mode can efficiently represent two regions contained in a MB, which can be vertically, horizontally or diagonally divided. However, this is not very efficient for arbitrary shapes. In this method, each segment of the MB arbitrary region is approximated by constant depth values. Using this approach, the author reported BDBR savings in depth maps up to 29%.

In [75], the authors followed an approach similar to the one presented in [73, 74], where the MB being encoded is segmented into separate regions. The number of regions is defined by $k$, and for each region a different prediction scheme is used. Using 2 and 3 regions, respectively for $k = 2$ and $k = 3$, this method is able to achieve significant coding efficiency outperforming H.264/AVC intra-coding, and showing an improvement of aproximately 6%(BDBR) over the methods described in [73, 74].

Besides the significant improvements of the methods described in [71, 73–75] over H.264/AVC intra prediction, the use of transforms to encode the residual information of depth maps results in significant distortion, especially at high frequency components. This has motivated further research to find new coding methods based on different paradigms, in order to preserve sharp edges of depth maps, obtaining synthesised views with higher quality. For example, in [76] Platelet encoding method is proposed, based on piecewise-linear functions [77] that attempt to approximate the shape of the depth content using linear functions and a quadtree decomposition [78]. Each depth map is hierarchically decomposed by division into smaller blocks, which are modelled by a function. To avoid using a large number of small blocks along depth discontinuities, the corresponding region boundary is defined with a straight line, and each one of these regions is encoded using an independent function. The result of using this technique is an improved rendering quality, when compared to H.264/AVC intra mode, mainly due to the geometric depth map feature (edges) that are better preserved.

The authors in [79–81], presented DC coding methods that have several aspects in common with the highly flexible partition scheme of Multidimensional Multiscale Parser (MMP) coding [82]. In MMP, the intra prediction modes are similar to those of H.264/AVC, but the residual information is encoded using a pattern matching technique, based on an adaptive dictionary. When this method is used to encode depth maps, it is shown that using large MB sizes combined with the highly flexible MMP partitioning, the coding efficiency if highly improved [80]. This is mainly due to the use of large blocks, which are suitable to represent homogeneous areas, and the flexible partioning allows to preserve depth map shapes, resulting in synthesised views with high quality.

In order to improve the overall quality of the synthesised views, for depth map coding, View Synthesis Optimisation (VSO) can be used [83–87]. This method uses a rate-distortion (RD) optimization, to achieve the best relation between the compressed depth maps and the synthesised views. For instance, the VSO technique described by the authors in [86], it is considered the exact regions on the synthesised image of the corresponding depth map being encoded. The depth coding is optimised based on a distortion metric that uses the synthesised regions using as a reference the original depth data and the corresponding synthesised regions. The authors reported a significant improvement in the depth coding efficiency by achieving bit-rate savings up to 17% using the proposed VSO technique.

# 3 Error concealment techniques

This chapter presents a review of the most common error concealment (EC) techniques, used in 2D and 3D/multiview video. In fact, most of the EC methods proposed for 3D image/video applications are based on techniques used for 2D video.

This chapter covers EC techniques used at the decoder side, classified as *Post Processing*, that is the main topic of this thesis.

## 3.1 Overview

EC methods can be classified into three different categories, depending on the role of the encoder and the decoder: *Forward EC*, *Interactive EC* and *Post Processing EC*.

*Forward EC* methods add redundancy to the video data in the encoding process, in order to increase the bitstream error resiliency. Some example of these type of techniques include layered coding with prioritized transport, using scalable video coding (SVC). SVC is implemented in the H.264/AVC standard [88] and is also under development for the H.265/HEVC standard [89, 90]. It allows to encode the video signal in different layers that are complementary between each other, which can increase error resiliency or the decoder flexibility. Multiple Description Coding (MDC) [91] is also an effective way of increasing the bitstream error resiliency, since the video signal is encoded using multiple

representations (or descriptions) that are fully independent and decodable, but the maximum quality is only achieved when all descriptions are received. MDC schemes allows to decode one single description when all others are lost in the network resulting in a decoded signal with a lower quality than that obtained from all descriptions. *Transport level control techniques* also fit into this category of methods. They rely on robust packetization schemes capable of recovering important information in presence of packet loss, such as coding modes and headers that are embedded in successive decoded packets, resulting in bitstreams with higher error resiliency [92]. Interleaving packets is also a transport level mechanism that is very efficient for minimising the effect of error bursts. When considering interleaved video packets, errors are spread over time thus improving the efficiency of the concealment process [93, 94]. For instance, authors in [95] use the H.264/AVC Flexible Macroblock Ordering (FMO) tool [93, 96] in the source signal, spreading errors along the video sequence. When bursts occur, they exploit FMO tool with a data hiding embedded technique, where edge information of each macroblock (MB) is embedded into the next frame.

The second category, *Interactive EC*, is based on the assumption that the sender (encoder) and the receiver (decoder) have a back channel, used to communicate between the decoder and encoder. Then, the encoder may have an active role in the concealment process, based on the decoder feedback information. Thus, coding parameters can be adjusted in order to dynamically improve the concealment process at the decoder (e.g. intra refresh). At the transport level, the feedback information can also be used to adjust the bandwidth capacity and availability employed in both video transmission and information feedback [97].

*Post Processing* techniques are only used at the decoder side. Since these are the main topic of this thesis, they will be described with detail in Sections 3.2 and 3.3. The first section describes *Spatial EC* methods, while the second section discusses *Temporal EC* methods.

## 3.2 Spatial Error Concealment for 2D video

In video communication systems, spatial EC methods play a very important role in intra-coded images due to the special role of these type of images in stopping temporal error

propagation. These type of methods allow to recover regions in corrupted frames using information only from their spatial neighbourhood, which is particularly important for images encoded without temporal dependencies like intra-coded, where motion information is not available. In such cases, these images can effectively stop the error propagation along the *Group Of Pictures* (GOP). Usually this is accomplished by I (Intra coded) pictures and IDR (Instantaneous Decoder Refresh). Among others, this type of pictures has the important role of preventing error propagation through different GOPs, thus the GOP size is directly related to the maximum period of time over which a single error can be visible. Common GOP sizes correspond to around half or one second of video. Therefore, efficient spatial error concealment in I frames combined with an adequate GOP size are crucial factors to limit the error propagation along a GOP, and to contribute for significant quality improvement of decoded video quality in the presence of transmission errors.

In this section, the most common spatial EC methods applied to 2D video are described. The goal of these methods is to recover corrupted regions that appear in decoded images due to losses during transmission. In general, lost regions corresponds to blocks of $16 \times 16$ pixels, or multiples, due to the block size used in video coding standards like MPEG-2 and H.264/AVC. In H.265/HEVC the coding unit (CU) size is $64 \times 64$ pixels. As expected, lost regions comprised of smaller size blocks are more efficiently recovered, due to the richer information in the neighbourhood. The class of Spatial EC methods can be divided into two types: *Spatial domain interpolation* and *Frequency domain interpolation*.

### 3.2.1 EC-Spatial domain interpolation

Spatial domain interpolation techniques rely on the availability of correctly decoded pixels around the lost region. MB copy is one of the simplest and less complex spatial EC methods [98]. In this method, the lost MBs are interpolated by simply copying the neighbouring pixels from the correctly decoded MBs. This method has a major drawback related to the mirror-like artefacts that can appear in the recovered regions.

Another spatial EC technique commonly used in H.264/AVC is based on weighted interpolation of the adjacent and undamaged neighbour values, which has been used in some research works [99–103]. Despite the good recovering performance in smooth regions and low computational complexity, this EC method tends to blur edges, adding distortion in

high frequency textures. In this method, a pixel ($Y'_{(x,y)}$) being recovered is interpolated by using four neighbour pixels located outside the lost region, as shown in Figure 3.1, where $b_{size}$ corresponds to the pixel size of a square lost region $y_j = [y_1, y_2, y_3, y_4]$.

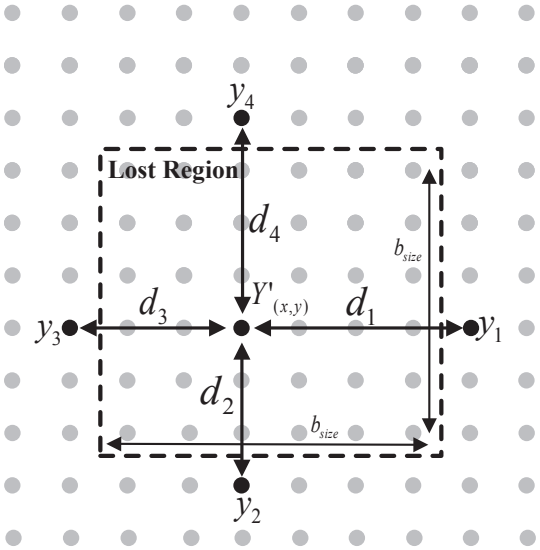



**Figure 3.1** – Spatial interpolation using weighted interpolation.

This operation is defined by Equation 3.1,

$$Y'_{(x,y)} = \frac{\sum_{j=1}^{N} y_j \times (b_{size} - 1 - d_{j\rightarrow(x,y)})}{\sum_{j=1}^{N} (b_{size} - 1 - d_{j\rightarrow(x,y)})}, \quad N = 4, \quad (x,y) \in LostRegion \tag{3.1}$$

where a pixel $Y'_{(x,y)}$, with coordinates $x$ and $y$, is the interpolated one, $j$ is the number ($j \epsilon [1,4]$) of each pixel sample $y_j$ used for interpolation, $d_{(j)\rightarrow(x,y)}$ is the distance between spatial coordinates $(x,y)$ and the pixel sample $(y_j)$. The inversely related of this distance, given by $b_{size-1} - d_{j\rightarrow(x,y)}$ is used as a weight for $y_j$, in order to give more weight to pixels that are closer to the one being concealed.

There are many techniques based on this approach, that besides using the error-free neighbour pixels, also take into account geometric information of the scene such as edges of visual objects, due to their importance in perceptual quality [104]. Since blurred or distorted edges caused by wrongly recovered pixels seriously affect the quality of experience, some EC concealment techniques were proposed to adequately deal with edges in video content in order to achieve improved results. This is reported in several previous research works, namely in [99, 103, 105–109], where edge information is analysed and extracted

from the correctly decoded regions to recover the lost ones. For instance, some authors use the well-known *Sobel* edge detector [110] to extract edge information [99, 105, 107–109, 111]. When MBs surrounding a corrupted area are available, a gradient filter can be applied, as illustrated in Figure 3.2, by convolving the horizontal and vertical *Sobel* masks with the correctly decoded neighbour area $F_{i,j}$ around the lost region, where $i$ and $j$ are the respective pixel coordinates. *Sobel* masks are defined by Equations 3.2 and 3.3.

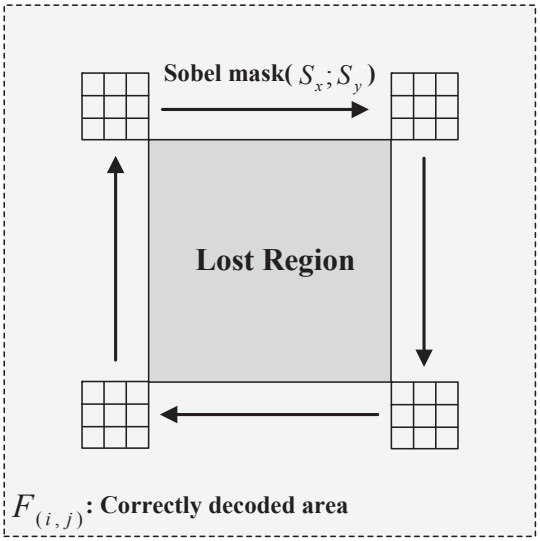



**Figure 3.2** – Aplying the *Sobel* mask in the lost region.

$$Sx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.2}$$

$$Sy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.3}$$

The gradient located at the pixel $(i, j)$ is defined by $G_{(i,j)}$, as shown in Equation 3.4.

$$G_{(i,j)} = (Gx_{(i,j)}, Gy_{(i,j)}) \tag{3.4}$$

$Gx$ and $Gy$, the horizontal and vertical gradient components, respectively, are defined by convolution operations, as shown in Equations 3.5 and 3.6.

$$Gx_{(i,j)} = F_{i,j} * Sx_{(i,j)} \quad (3.5)$$

$$Gy_{(i,j)} = F_{i,j} * Sy_{(i,j)} \quad (3.6)$$

The gradient magnitude at the pixel $(i, j)$ is represented by $MG_{(i,j)}$ and defined by Equation 3.7, while the angle of each edge point is defined by Equation 3.8.

$$MG_{(i,j)} = \sqrt{Gx^2_{(i,j)} + Gy^2_{(i,j)}} \quad (3.7)$$

$$\alpha_{(i,j)} = \arctan\left[\frac{Gy_{(i,j)}}{Gx_{(i,j)}}\right] \quad (3.8)$$

If a certain value $MG_{(i,j)}$ is larger than a predetermined threshold, then it is considered an edge point, and can be represented in a binary map defined by $Ed_{(i,j)}$. The threshold definition is very important since it affects substantially the performance of the edge detection, and consequently the concealment performance. For the sake of simplicity, many implementations described in the literature rely on thresholds based on the variance of the image regions where edge detection is being performed. This is a simple way of defining the threshold but might not be the best option if the image is not homogeneous, because it may result in false edges. To minimize this problem, other edge detection techniques are also used, such as *Canny* edge detection [112, 113], where the image is firstly smoothed in order to remove noise and consequently reduce false edge detection. The *Canny* algorithm does not use a single threshold but two, in order to implement hysteresis thresholding. The use of two thresholds allows significant reduction of broken contours caused by magnitude fluctuation ($MG_{(i,j)}$) that could be close to the value of a single threshold.

After computing all edges in the surrounding area of the lost region, and the corresponding directions $\alpha_{(i,j)}$, the next step is to link the edge points that are estimated as belonging to a contour that is broken due to the occurrence of lost data. Finally, those edge end-points are linked in order to recover the interrupted contour.

Figure 3.3 shows an example the contour recovering in a lost region [99, 111]. The four small black squares aside the lost region are considered to be broken contour endpoints

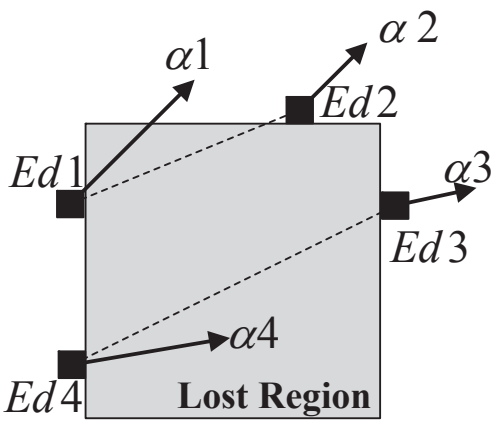



**Figure 3.3** – Example of contour edge endpoints matching and linking.

($Ed1$ to $Ed4$) and angles $\alpha 1$ to $\alpha 4$ correspond to the angles of contours entering in the missing region. By determining these angles, it is possible to establish a relation between endpoints, considering that the contours that point to each other were probably connected. Finally, they are linked using a straight line, as exemplified by the dashed-line. This type of approach works satisfactorily for small lost regions when the objects in the scene have well-defined and straight contours. In the case of corrupted curved shapes, this type of approach has the tendency to introduce undesirable artefacts, resulting in inaccurate recovered contours.

After recovering the corrupted contours, the next step is to interpolate pixels of the missing region in a similar manner, as defined by Equation 3.1. The main difference is that the selected pixels for weighted interpolation are now chosen taking into account the recovered contours based on edge directions. After delimiting regions by contours, the weighted interpolation is performed using the correctly decoded neighbours, but only using pixels belonging to the same region, as illustrated in example of Figure 3.4. Depending on how the recovered edges surround the pixel to be recovered, the number of neighbour used pixels might be different for each case. In some cases only one neighbour is used, as shown in of Figure 3.4 c) for $p_2$.

The authors in [114] proposed an EC method to recover MBs in H.264/AVC decoders, simulating losses based on the assumption that FMO was used by the encoder for improved resilience of aH.264/AVC. In this case when a slice group is lost, the missing regions result

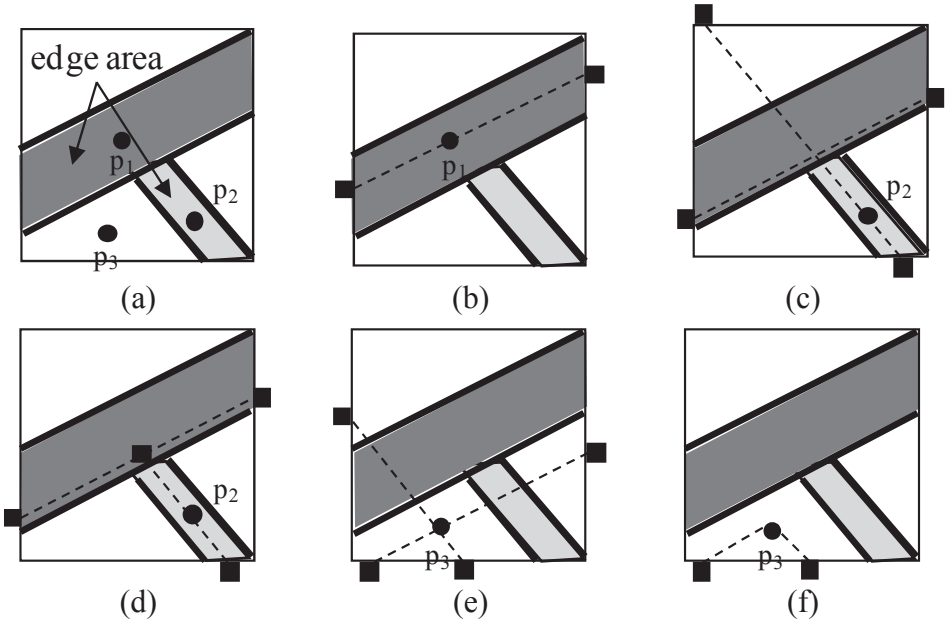



**Figure 3.4** – Six different cases for weighted interpolation pixel selection[99].

in a chess-like error pattern dispersed through the whole image. This technique is divided into two main steps: *Block Classification/Directional Decision* and *Block Concealment.*

In the first step, the edges are extracted along with their directions using the correctly decoded neighbour MBs. The approach to extract this information is very similar to the one previously described, where $3 \times 3$ vertical and horizontal *Sobel* operators are used. If all neighbour blocks of the lost MB are considered to be homogeneous, due to the absence of edges, the lost MB is reconstructed using weighted averaging, as defined in Equation 3.1. In the presence of edges, the corresponding angles are approximated in order to match the eight possible directions of H.264/AVC Intra mode, as shown in Figure 3.5 for $4 \times 4$ blocks [115].

The second step is where the main novelty of this technique resides. The lost MB is split into $4 \times 4$ sub-blocks and the goal is to recover the intra prediction modes based on the computed directions of the first step. Based on these directions, eight prediction modes can be used: vertical, horizontal, diagonal down/left, diagonal down/right, vertical right, vertical left, horizontal down and horizontal-up. Based on the estimation of these eight prediction modes, the lost $4 \times 4$ blocks are interpolated by using the neighbour samples, which are represented by the top (Q, A, B, C, D, E, F, G, H) and left (Q, I, J, K, L)

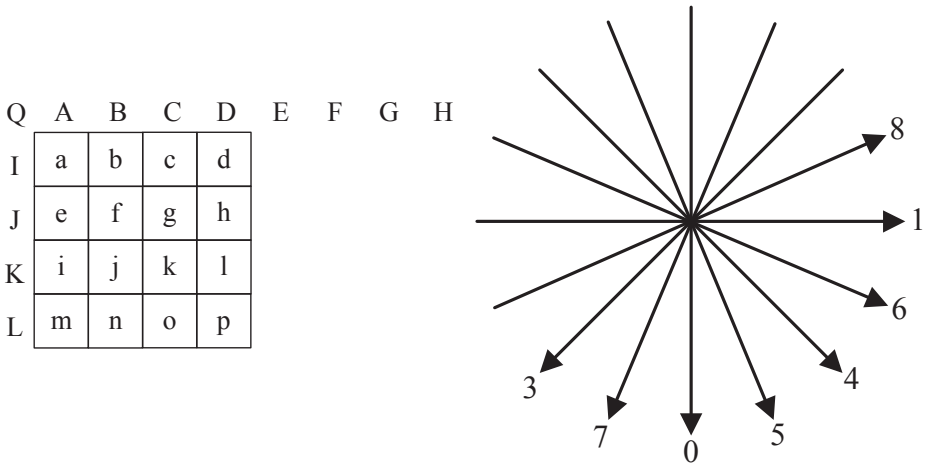



**Figure 3.5** – H.264/AVC intra prediction directions.

letters, as shown in Figure 3.5. This method achieves good results, when compared to the reference ones *weighted pixel averaging* and *multi-directional-interpolation* [105], because the edges are recovered more efficiently.

In general, in spatial EC techniques, strong edges passing through a lost region can be recovered quite effectively, but in regions with higher texture details the reconstructed area is often blurred. Besides the reasonable concealment performance of these methods, they also present low complexity, which is an important feature for any EC technique.

### 3.2.2 EC-Frequency domain interpolation

Frequency domain EC methods are also based on correctly decoded neighbour areas, but the recovery process is intended to reconstruct the corrupted DCT coefficients. Relevant techniques described in the literature are discussed in the following.

X. Lee in [116, 117] proposed a method that relies on fuzzy logic reasoning to recover high frequency information of lost blocks. An EC in the pixel domain is first performed (similarly as the methods described in Section 3.2.1)on the smooth areas and the high frequency details are refined using fuzzy logic. The accuracy, highly depends on the size of the lost regions and the larger they are, the more difficult is to recover the high frequency components, and consequently the EC is less efficient. The algorithm comprises three steps. Initially a pixel domain EC of the lost area is performed. Then, lost blocks

containing high frequency components are recovered by using Fuzzy logic reasoning. Finally, a sliding window technique is used to combine the outputs given by the previous two steps.

In the first step, *Hierarchical Compass Interpolation/Extrapolation* (*HCIE*) is performed assuming 2D image continuity is assumed, which is valid for lost regions where low frequency components are dominant. Homogeneous and low frequency areas are recovered by first estimating the global edge orientation from eight DC-coefficient components of the surrounding blocks using compass-edge operators. These compass operators are based on the principle described in previous Section 3.2.1 that used Sobel operators. In this case, the authors use eight Sobel operators to compute, not only the vertical and horizontal directions, but also six more intermediate directions. These eight directions are named $South$, $S-east$, $West$, $N-West$, $Vertical$, $L-diagonal$, $Horizontal$, $R-diagonal$. After estimating the edge directions, a cubic spline interpolation is performed along estimated edges in order to link the edge points. The final task of this first step is to reconstruct the lost pixels by using an extrapolation of the recovered edges and the correctly decoded neighbour values. This method works well on small blocks and when there is a small amount of high frequency components. This step tends to exhibit similar behaviour as a low pass filter because complex textures with high frequency content tend to disappear.

In the second step, a Fuzzy logic approach is used in the DCT domain, in order to recover the lost coefficients. Since most transform-based encoders rely on the DCT and their coefficients are available at the decoder, a computation step can be avoided, increasing the simplicity of the algorithm. In the DCT domain, using $N \times N$ sized block, there is one DC coefficient and $(N \times N) - 1$ AC-coefficients. The authors divided these coefficients into six categories: DC, low frequency, horizontal texture, vertical texture, diagonal texture and high frequency texture, as shown in Figure 3.6 for a blocksize of $8 \times 8$, containing 64 coefficients. In this example $C_i, i = 0$ is the DC-coefficient component and the other are the AC ($i = 1, 2 \ldots 63$).

A set of four features is used to interpret the spectral information of the DCT coefficients: the energy distribution within each sub-spectra, the energy gravity centre of each sub-spectra, the texture orientation of the lost block and, finally, the phase template of each

| DC | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 2 | 3 | 3 | 3 | 3 | 1 | 1 |
| 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 |
| 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 |
| 2 | 2 | 3 | 3 | 4 | 4 | 4 | 4 |
| 2 | 2 | 2 | 4 | 4 | 4 | 4 | 4 |
| 2 | 2 | 2 | 4 | 4 | 4 | 4 | 4 |

AC coefficents:
0-Low frequency.
1-Horizontal texture.
2-Vertical texture.
3-Diagonal texture.
4-High frequency.

**Figure 3.6** – DCT coefficients categorization for a $8 \times 8$ block.

sub-spectra. The normalized energy of each sub-spectra is defined by Equation 3.9:

$$E_k = \frac{\sum\limits_{i \epsilon \psi_k} C_i^2}{\sum\limits_{j \epsilon \Omega} C_j^2} \tag{3.9}$$

where the subset $\psi_k, k = 1, 2, \ldots 5$ represent, respectively the low frequency, horizontal, vertical, diagonal and high frequency sub-spectral content. $\Omega$ represents the subset of all coefficients except the DC and the first five AC low frequency. For the blocks with simple texture patterns, it is assumed that correctly decoded neighbour blocks have similar features and also similar orientations. Based on this assumption, it is considered that the lost region has similar characteristics and, consequently, with an analogous energy distribution. Fuzzy logic is used to recover the lost coefficients based on the texture patterns of the correctly decoded neighbours [116, 117]. The neighbours with the highest sub-spectra energy give the orientation of the texture block being recovered. The final step combines the reconstructed data obtained from the spatial domain and frequency domain techniques using a sliding window technique. As mentioned earlier, the spatial domain *HCIE* is used to recover the lower frequency domain areas, and Fuzzy logic is used to recover higher frequency components. A sliding window iteration (SWI) is used to perform a boundary analysis, which combines the optimal results between *HCIE* method and the frequency domain Fuzzy inference conditions. To simulate the lost regions, the authors considered an error pattern where small $8 \times 8$ blocks are lost, equally dispersed through the entire image. The method provides good results in the proposed scenario,

but simulations were made using only small blocks losses.

In [118–120] an EC method is proposed based on 2D DFT basis functions, in order to extrapolate the missing data at the corrupted regions. DFT, as well as DCT, are suitable transformation techniques to extrapolate missing image coefficients because they imply that the original function is periodic. In contrast, polynomials and wavelets might not the most suitable when the missing area is large. DFT and DCT are also quite effective to restore missing regions that contained textured areas with high frequency content, when this content has periodic characteristics. The basis functions are selected according on the corrupted image characteristics. This EC method uses a *Discrete Linear Approximation* based on the *Principle of Successive Approximations*. The authors used an approach based on a coding technique for segmented images as described in [121], but applied in this case to an EC algorithm. The texture is successively approximated and then segmented to the shape of the object. In this EC method, the missing data is extrapolated by successive iterations with the support of the surrounding area.

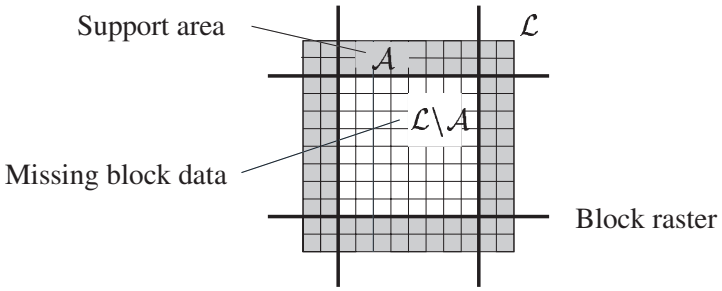



**Figure 3.7** – Missing region and respective support area [118].

An example of a missing region is illustrated in Figure 3.7 in white colour ($\mathcal{L}\backslash\mathcal{A}$), a support area is represented in grey ($\mathcal{A}$) and the entire image as $\mathcal{L}$. The region $\mathcal{A}$ is approximated by a successive DFT basis functions representing the entire area $\mathcal{L}$. Each function can provide estimation for the missing coefficients, but all functions are used together, adopting a weighted linear combination. The complexity of this method depends on the type of the images. If the region being concealed has homogenous characteristics, then one iteration might be enough. In the presence of more textured regions, more iterations are needed, which naturally increases the EC complexity. Also in this case it is assumed that uncorrupted areas exist around the missing regions. Despite the good results reported, random error patterns along the images were not considered, which limits

the validity of the results when considering real applications, where the loss can occur anywhere. Observing the recovered images provided by the authors, the strong edges seem to be efficiently recovered, but in areas highly detailed, this EC method has a behaviour similar to a low-pass filter.

## 3.3 Temporal Error concealment

The temporal redundancy between adjacent frames has also been exploited by temporal error concealment techniques (TEC) to recover from transmission errors. In the last generation of video standards since MPEG-1, three types of frames are defined according to their coding type: Intra coded frames (I), Predictive-frames (P) and bi-predictive-frames (B). In most coding configurations, bi-predictive frames, which contain predictions from past and future frames, are not commonly used as reference pictures. Thus, if a B frame is lost or corrupted during transmission, it will not affect subsequent frames, regardless the concealment method used to recover it. In the case of I and P frames, this must be dealt more carefuly because even small localized errors can be strongly propagated through the whole GOP, affecting many other frames. Nevertheless, depending on the coding modes and transmission scenarios, the effect of error propagation is not easily predictable and the best concealment methods are those capable of dynamically adapting their operation to the time-varying transmission conditions of each specific transmission infrastructure.

Temporal error concealment methods can be classified into the following categories, from the simplest to the most complex: Frame Copy (FC) or Temporal Replacement (TR), Motion Copy (MoCp), Motion Boundary Match Algorithms (MBMA) and Motion Vector Extrapolation (MVE). The following subsections present a short review on these techniques.

### 3.3.1 Frame Copy (FC)

FC is one of the temporal EC methods with lower complexity, which makes it almost a default method in many practical video decoders. By simply copying the nearest frame

decoded without errors or only co-located pixels of the lost region, FC is equivalent to zero-order interpolation in the temporal domain. FC is also known as Temporal Replacement (TR), since all motion information about the lost regions is simply replaced by zero MVs. Using such an EC method only provides good results in very low motion video, otherwise it results in the well known effect of frame freezing.

### 3.3.2 Motion Copy (MoCp)

MoCp is a simple EC algorithm slightly more complex than FC, but still requiring fairly low complexity in most straightforward implementations [122, 123]. This type of method is suitable for entire frame losses, because only motion information from previously decoded frames is used. MoCp performs motion compensation from the nearest decoded frame to recover the corrupted area. Typically this method is based on the assumption that motion along a video sequence is mostly translational with constant velocity, thus temporally scaled MVs from nearest correctly decoded frame may be useful to reconstruct corrupted frames.

Figure 3.8 shows an example of a current lost MB (*Curr. MB N*) that is being concealed in frame $n$. Assuming that frames $n-1$ and $n-2$ were decoded without errors and only one reference frame is used, MVs from frame $n-1$, which are referenced to frame $n-2$, can be just copied from $n-1$ to $n$. Then they are used to recover the lost frame through motion compensation.

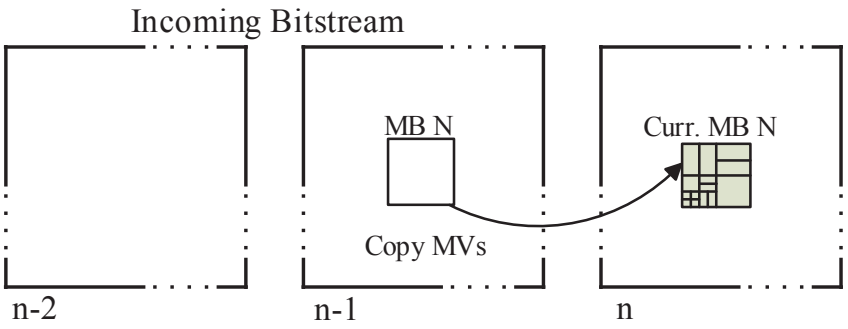



**Figure 3.8** – Example of Motion Copy(MoCp) EC method.

This method has good performance when the motion vectors (MV) characteristics represent homogeneous motion, not changing significantly over time. When the objects in the

scene have complex motion characteristics, such as fast moving objects, rotations, accelerations, among others, the use of such motion vectors will result in poor EC performance.

### 3.3.3 Motion Boundary Match Algorithms (MBMA)

Motion Boundary Match Algorithms (MBMA) are also often called simply BMA. This type of algorithms are used when only some parts of the frame are lost and there exist some correctly decoded areas around the lost regions.

MBMA techniques can use the correctly decoded neighbour regions using two different types of information: temporal and spatial information. Relying on the assumption that lost regions are highly correlated with the correctly neighbour regions, the available neighbour MVs are used to conceal the lost regions. Additionally, motion information extracted from the neighbour blocks can be further refined using a matching algorithm at the borders of lost areas. There are different approaches to implement the boundary matching method. In some methods, boundary matching is performed by computing the distortion between the pixels located inside the boundaries of the lost MB and the pixels right outside the border of the lost region. Another popular approach is to use a support area, which is an overlapping region at the outside boundaries of the recovered block. In both approaches, the main objective is the same: to find the candidate block that produces the smallest possible distortion at its boundaries.

Figure 3.9 shows an example of how the boundary block match is typically performed. Considering a corrupted MB, where the left, top, right and bottom MBs are available, a support area of N pixels is used in those regions. The vertical and horizontal coordinates are $x$ and $y$, respectively, and the origin of the coordinate system ($(x, y) = (0, 0)$) is located at the first pixel of the MB. $N$ corresponds to the rows (top and bottom MB side) or columns (left and right MB side) of the candidate block that overlap at the outside of the corrupted region. In this example, the sum of absolute differences (SAD) is used as distortion metric, but mean square error (MSE), mean of absolute differences (MAD) among others can also be used.

To recover the lost MB with $H \times V$ size, the candidate block must be $(H+2N) \times (V+2N)$ is used. The SAD for the four overlapping regions ($SAD_{TOT}$) is defined by Equation 3.10.

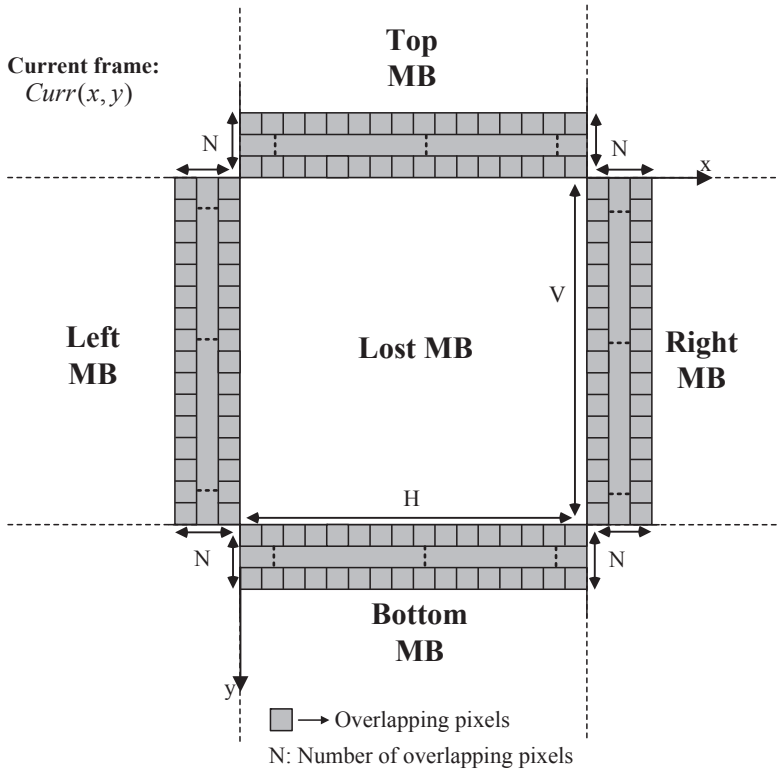



**Figure 3.9** – Example of a typical boundary matching process.

$$SAD_{TOT} = SAD_{LR} + SAD_{TB} \tag{3.10}$$

$$SAD_{LR} = \sum_{\substack{0 \leqslant x \leqslant N \\ 0 \leqslant y \leqslant V}} |Curr_{(-x,y)} - Cand_{(x,y)}| + |Curr_{(x+H,y)} - Cand_{(x+H,y)}| \tag{3.11}$$

$$SAD_{TB} = \sum_{\substack{0 \leqslant x \leqslant H \\ 0 \leqslant y \leqslant N}} |Curr_{(x,-y)} - Cand_{(x,-y)}| + |Curr_{(x,y+V)} - Cand_{(x,y+V)}| \tag{3.12}$$

The individual SAD values for the respective left/right and top/bottom areas are defined by Equations 3.11 and 3.12. $Curr(x, y)$ represents the current corrupted frame and $Cand(x, y)$ is the candidate block to reconstruct the lost MB. For each MV candidate there is a candidate block ($Ca$) and a corresponding $SAD_{TOT}$. The best MV is the one with the smaller $SAD_{TOT}$.

Two EC algorithms were presented in [124] based on boundary matching algorithms. The first proposes the reconstruction of the lost MVs using bilinear interpolation of the correctly decoded MVs while the second additionally combines a boundary-matching algorithm. The first EC method, called Bilinear Motion Field Interpolation (BMFI), assumes that four neighbour MVs from top left ($V_{TL}$), top right ($V_{TR}$), bottom left ($V_{TR}$) and bottom right ($V_{TR}$) of the corners of the lost block are available. The interpolated MV used to recover the lost region is represented by $v(x, y)$ at a given point $p(x, y)$, and it is computed using bilinear interpolation, as defined by Equations 3.13 and 3.14.

$$v(x,y) = (1-x_n)(1-y_n)V_{TL} + x_n(1-y_n)V_{TR} + (1-x_n)y_nV_{BL} + x_ny_nV_{BR} \tag{3.13}$$

$$x_n = \frac{x - x_L}{x_R - x_L}\ ,\ y_n = \frac{y - y_T}{y_B - y_T} \tag{3.14}$$

$x_L$ and $x_R$ are, respectively, the abscissa coordinate of the left and right borders of the lost block, while $y_T$ and $y_B$ are the coordinates of the top and bottom borders, respectively. It is assumed that motion information of neighbour blocks is available and the lost block is recovered using motion compensation with the MVs computed by Equation 3.13.

The second EC method also uses BMFI combined with other technique, named Boundary Matching algorithm with Side Distortion Measure (BMSMD) [125, 126]. Since BMFI use MVs from the neighbouring regions, the motion field computed for the missing regions tends to be smooth. In some cases this is an attractive feature, but in some others this technique fails if the local motion characteristics are not highly correlated to its neighbours. BMFI is also extremely dependent on the availability of the neighbouring regions. In the case of reduced availability of neighbours MVs, the performance will considerably decrease. BMSMD is not so highly dependent on the availability of the neighbour MVs as BMSD because in this method only the MVs that minimise the distortion at the boarders of the lost region are individually used. The authors combine the two methods using Equation 3.15.

$$\hat{p}_c(x,y) = \frac{1}{2}\left[\hat{p}_r(x + \hat{d}_x^{\,i}, y + \hat{d}_y^{\,i}) + \hat{p}_r(x + \hat{d}_x^{\,s}, y + \hat{d}_y^{\,s})\right] \tag{3.15}$$

where $\hat{p}_c(x,y)$ are the recovered pixels at coordinates $(x,y)$ and $\hat{p}_r$ refers pixels from the reference frame. MVs obtained by using BMFI and BMSMD are defined, respectively, by $\hat{v}_c{}^i(x,y) = (\hat{d}_x{}^i, \hat{v}_y{}^i)$ and $\hat{v}_c{}^s(x,y) = (\hat{d}_x{}^s, \hat{v}_y{}^s)$ where $(\hat{d}_x{}^i, \hat{v}_y{}^i)$ and $(\hat{d}_x{}^s, \hat{v}_y{}^s)$ are the MV vertical and horizontal components, respectively. Concealment of the lost region using the combination of both methods, as defined by Equation 3.15, is in practice the average of two motion compensated blocks using BMFI and BMSMD MVs. The authors used the Peak Signal to Noise Ratio (PSNR) objective quality metric, as defined by Equation 3.16. $MAX$ is the maximum possible pixel value of the images while the MSE is the Mean Square Error as defined by Equation 3.17. $h$ and $l$ are the horizontal and vertical image resolution, respectively, $i$ and $j$ are the horizontal and vertical coordinates, respectively. $R$ is the reference image and $Im$ is the image under evaluation.

$$PSNR = 10log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{3.16}$$

$$MSE = \frac{1}{h \times l}\sum_{i=0}^{h-1}\sum_{j=0}^{l-1}[R(i,j) - Im(i,j)] \tag{3.17}$$

The authors reported a performance gain in PSNR approximately between 0.8db and 1dB, for a block loss rate of 20%, when compared to the case where BMFI and BMSMD are independently used. When compared to other popular concealment methods, such as *Temporal Replacement* (Frame copy) and motion vector averaging (AV), the PSNR improvements can be up to 2dB.

The authors in [127] proposed an adaptive EC algorithm to be used with H.264/AVC, that selects different concealment modes, depending on the MV range of the spatial and temporal neighbour blocks. Based on the local characteristics of the corrupted region, the EC mode can be dynamically changed. The matching criterion for the best MV is optimized using a boundary matching technique.

Since this EC method is based on H.264/AVC, it is based on the block partition modes of corrupted neighbour blocks to provide useful information to the proposed switching mode. Figure 3.10 shows the neighbour blocks that are used in the error concealment process, based on the similarity between the $LostMB$ and neighbour $NB_i$, where $i$ ndentifies the neighbour block.

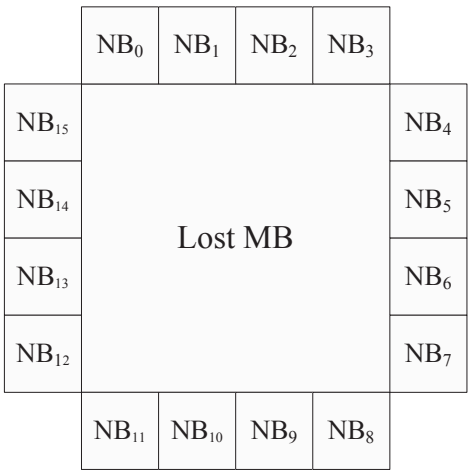



**Figure 3.10** – Concealment using $4 \times 4$ neighbours blocks [127].

Equation 3.18 defines the mode $m_i$ for the neighbour block $NB_i$. If the sum of $m_i$ for $i = 0, ..., 15$ is larger than 8, then the lost MB is recovered by dividing it into $8 \times 8$ sub-blocks. If the sum of $m_i$ is smaller than 8, then each lost MB is recovered as a whole.

$$m_i = \begin{cases} 1, & \text{if } Mmode_i = 8 \times 8,\ 8 \times 4,\ 4 \times 8,\ 4 \times 4 \\ 1, & \text{if } NB_i \text{ is unvailable,} \qquad \text{i=0,...,15} \\ 0, & else \end{cases} \tag{3.18}$$

After selecting the error concealment mode, the MV candidate is chosen. MBMA techniques usually use a set of four nearest neighbour MVs, located at the corners of the lost region, as described in [128]. In order to improve efficiency, the number of MV candidates is increased. Besides the spatially neighbour MVs, other MVs from the co-located MB of the previously decoded frame are also used. For each of these candidates, MV refinement is performed in order to find the best possible match. The authors use two criteria to perform the boundary matching process. The first is the smoothness of recovered pixels at the boundaries and the second criterion is motion uniformity. Based on these assumptions, the best candidate block to reconstruct the missing one is chosen. The authors claim good concealment performance, achieving even a small decrease in computational complexity, when compared to other EC techniques that also use a similar approach, such as the example presented in [128]. The Y-PSNR gains in the reconstructed sequences can be up to 0.91dB, when compared to the best reference method used by the authors.

In [129] another example of an EC algorithm for H.264/AVC that uses FMO error resilience tool is presented. The authors used FMO dispersed mode with two slice groups. In the experiments, only one slice group is lost for each frame and bursts of errors are not taken under consideration. The authors proposed a combined EC algorithm that switches between two methods based on a spatial evaluation criteria: the first one is a conventional temporal EC method and the second one, the proposed method which follows a set of five different steps:

Firstly, a conventional EC method is applied where a boundary matching technique is used, similar to the one described in Figure 3.9. Due to the use of FMO (dispersed mode with only two slice groups) if one of the slices is lost, then the diagonal MBs of the corrupted MBs are never available and only the top/bottom and left/right neighbours are correctly decoded. The method considers an overlapped region of four pixels ($N = 4$), thus in the optimal scenario where the four neighbour MBs are available, the number of overlapped pixels used in matching process is 256.

The second step is based on the best block found in the previous step, by computing the residuals at the boundaries of the recovered block. The same set of overlapped pixels as the ones used in the previous step for the boundary matching process are used to compute the residual values, resulting in 256 values. The computed residuals are then added to the recovered block, obtained in the previous step, in order to enhance the recovered region.

In the third step, standard deviation and temporal evaluation criteria are computed to choose the concealment method. The standard deviation $\sigma$ as defined by Equation 3.19 and 3.20 is computed for the same correctly decoded neighbouring region as used in the previous step. $P(i, j)$ defines the pixel value of the correctly received neighboring boundary region, while $\mu$ defines the mean value of $M \times N$ boundary pixels.

The temporal information criteria ($BD$) is based on the sum of absolute differences between the pixels of the inner border of the candidate ($MB_{en}$) and the outer border of lost MB that belongs to the current frame being decoded.

$$\sigma = \sqrt{\frac{1}{N \times M - 1} \sum_{i=1}^{N} \sum_{j=1}^{N} (P(i, j) - \mu)^2} \tag{3.19}$$

$$\mu = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{N} P(i,j) \tag{3.20}$$

In the fourth step, a new set of candidate blocks is computed in order to enlarge the number of options to conceal the lost MB. This new set of candidate blocks is based on the weighted average of the two ones previously computed: replaced MB ($MB_{rep}$) and $MB_{en}$. Then, the a weight is given to $MB_{rep}$ and $MB_{en}$ is the inverse of each other, in order to give the opposite importance to $MB_{rep}$ and $MB_{en}$ resulting in another set of candidates ($MB_{opt\_aw}$).

In the fifth, and final step, the best candidate block is chosen based on the standard deviation $\sigma$ and also the temporal evaluation criteria computed on the third step, using a set of pre-defined thresholds the candidate from the set [$MB_{rep}$; $MB_{en}$; $MB_{opt\_aw}$].

In this proposed method, the authors reported good results when compared with the reference methods, achieving consistent gains over all the tested sequences. The proposed method reveals to be efficient, but in order to validate the results, more details should be given. The authors do not specify if they considered losses also on intra frames, or if they rule out the possibility of error bursts.

### Discussion

Besides using the correctly decoded neighbour motion information (spatially and temporally), MBMA EC methods also perform MV refinement by analysing the boundary matching of the candidate blocks in the corrupted regions. The additional performance gain comes at the cost of higher computational complexity, due to the refinement process that relies on block matching techniques. As usual, the performance of this type of EC methods is also highly dependent on how severe are the errors in both spatial and temporal domain. In the spatial domain, if large image areas are corrupted or lost, the EC performance is severely affected due to the limited neighbour information that is needed in boundary matching process. In the temporal domain, errors or incorrectly decoded areas will affect not only the corrupted frames itself, but also future frames that are referenced to the concealed data. Therefore, good concealment performance is necessary in order to

limit the negative effects of error propagation. Besides the EC techniques presented above, there are many other variations, including EC methods more suitable to deal with partial losses inside a frame, such as single MBs scattered along the frame, rows of MBs, etc. The efficiency of such methods depend on how effectively motion and spatial information is used together. The number of slices used for each frame also affects the concealment performance because when the number of slices is higher, in the occurrence of lost data, the lost areas will be more scattered throughout the corrupted frame. But, naturally, this increased error resiliency comes at a cost of coding efficiency. Concluding, most of these methods are based on an identical approach: find the best candidate MVs by using either block matching algorithms or refining the MVs using block matching techniques [130–133].

### 3.3.4 Motion Vector Extrapolation (MVE)

Motion Vector Extrapolation (MVE), as other EC techniques described in Section 3.3.1, 3.3.2 and 3.3.3 are based on temporal correlation between frames. In MoCp, error concealment is based on the assumption that motion has stationary characteristics in adjacent frames. In some cases like a camera panning, where objects have constant or very low motion, MoCp is an efficient EC method. In other cases where the motion characteristics are not so predictable, MoCp often tends to fail and motion information of previously decoded frames cannot simply be copied into the damaged area. In the MVE techniques described in this section, MVs from one or multiple past and future frames are used to extrapolate a MV that represents with good accuracy the lost motion information. Several variants of MVE techniques are capable of improving the MoCp algorithm at the cost of some extra complexity. MVE is also a suitable technique to recover full frame losses, since the extrapolated motion information is based on past and/or future frames.

The MVE method proposed in [134] for whole frame losses, each MB is first divided into four blocks and the MV of each block is then extrapolated from the previously decoded frames. Afterwards, the dependence of each MB is analysed in order to obtain an optimized estimation of the interpolated MV. After interpolating the lost MVs, the lost MB is then reconstructed using motion compensation. Figure 3.11 shows an example of the MV extrapolation process for a damaged MB (Frame N, $MB = 19$). In this example, only one reference frame is used and for a certain frame $N$ it is the previous

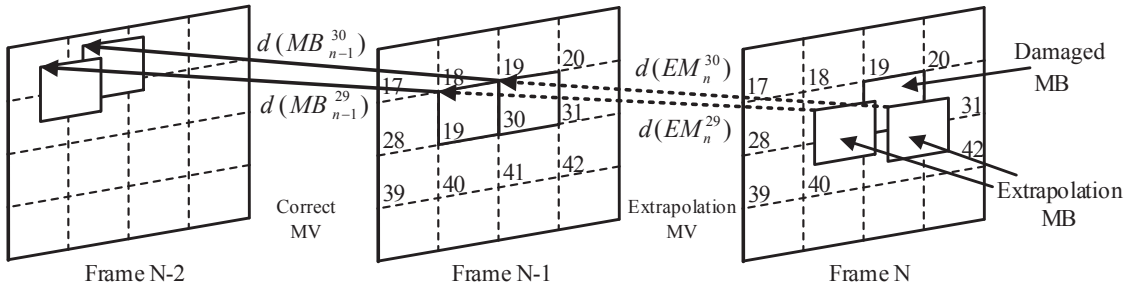



**Figure 3.11** – Motion vector extrapolation [134].

one, $N-1$. $MB^i_{n-1}$ represents MB $i$ belonging to frame $n-1$ while $d(MB^i_{n-1})$ is the respective MV that is related to frame $N-2$. $EB^i_n$ is the extrapolated MB based on $MB^i_{n-1}$, while $d(EB^i_n)$ is the corresponding extrapolated MV. $d(EB^i_n)$ is defined by a linear extrapolation, as defined by Equation 3.21.

$$d(EB^i_n) = d(MB^i_{n-1}), \quad i = 1, 2, ..., M \tag{3.21}$$

$M$ is the total number of MBs contained in one video frame. Considering a $MB^i_n$ to be recovered, this MB is then divided into four sub-blocks. These sub-blocks are defined as $B^i_{n,k}$, where $k = 1, ..., 4$ indicates the number of the sub-blocks, $d(B^i_{n,k})$ is the MV of $B^i_{n,k}$. Using the extrapolated MBs ($EM^i_n$) from frame $N-1$, the overlapped pixels with the MB to be recovered in frame $N$ ($MB^i_n$) are checked, as several extrapolated blocks $EM^i_n$ may be overlapped with $MB^i_n$. The MV of each sub-block is obtained by choosing the extrapolated MV $d(EB^i_n)$ that results in an MB with the highest similarity with $MB^i_n$. The comparison is measured by counting the number of overlapped pixels. The higher is this number, the higher will be the correlation between $EM^i_n$ and $B^i_{n,k}$. In the case where extrapolated blocks do not overlap the missing MB, the MV $d(B^i_{n,k})$ is obtained by using the nearest MV of the left block. If the left block does not contain motion information, also due to the lack of overlapped $EM^i_n$ or when the block is intra-coded, $d(B^i_{n,k})$ is set to zero, which is the same of temporal replacement EC method (TR).

When compared to the most common EC methods, such as MoCp, TR and MV averaging (AV), the authors have shown that the proposed MVE EC method is able to achieve PSNR gains over 1dB in comparison with the best EC reference method (AV). This method has the advantage of having a low computational complexity, because only the overlapped

extrapolated MVs are used for concealment and block matching is not performed to refine the interpolated MVs.

S. Belfiore, in [135] also proposed an EC method to reconstruct entire frames, lost in transmission networks. When compared to MVE, this method uses motion information not only from the previously decoded frame, but also from a wider group of frames that might be available. In the H.264/AVC and H.265/HEVC standards, up to sixteen references frames can be used, and the authors rely on this enlarged set of frames to analyse motion over a longer period of time, in order to achieve more effective interpolation of MVs to reconstruct the lost frame. In this method, after determining the number of past frames available in the reference buffers, these are indexed, considering that the lost frame is indexed by $n$, while the previous ones range from $n-1$ to $n-L$, where $L$ is the number of available decoded frames. An MV is defined by $MV_{i,j}^{n}$ for the respective pixel $x_{i,j}^{n}$, at location $(i, j)$. After the processing described above, the proposed method follows six main steps:

•**Step 1:** In the first step, an MV history is generated, computed from all frames from $n-1$ to $n-L$. In Figure 3.12 it is shown an example where this task is performed from $n-1$ to $n-3$ ($L=3$). $MVH_{i,j}^{n}d$ defines the MV history, starting from frame $n-1$, $d$ defines the horizontal ($d=h$) or the vertical ($d=v$) components of $MVH_{i,j}^{n}d$. For each pixel $x_{i,j}^{n}$, the MV $MV_{i,j}^{n-1}$ is stored into $MVH_{i,j}^{n-1}$, and the direction of MV pointing to $n-2$ is verified, while the corresponding MV $MV_{i,j}^{n-2}$ is stored into $MVH_{i,j}^{n-2}$. The MV history is generated until frame $n-L$ is reached, or until an intra-coded pixel is encountered or an MV point to an area which is out of bound of the respective frame.

•**Step 2:** In the second step, the MVs stored in the list are used to perform motion compensation on the pixels of frame $n-1$. It is assumed that each pixel from frame $n-1$ will move to those of the lost frame $x_{iF,jF}^{n}$. $iF$ and $jF$ are the final horizontal and vertical coordinates, respectively. $iF$ is defined by computing the average of the MVs present in the history list as defined by Equation 3.22

$$iF = i + \frac{1}{L_M}\sum_{k=1}^{L_M} MVH_{i,j}^{k}d, \quad d=h \tag{3.22}$$

$L_M$ is the number of MVs in the history of a pixel $i, j$. $jF$ is computed similarly to $iF$, but in this case, $d=v$. $iF$ and $jF$ are then used to define a motion field ($FMV$) that is defined by the following of Equations:

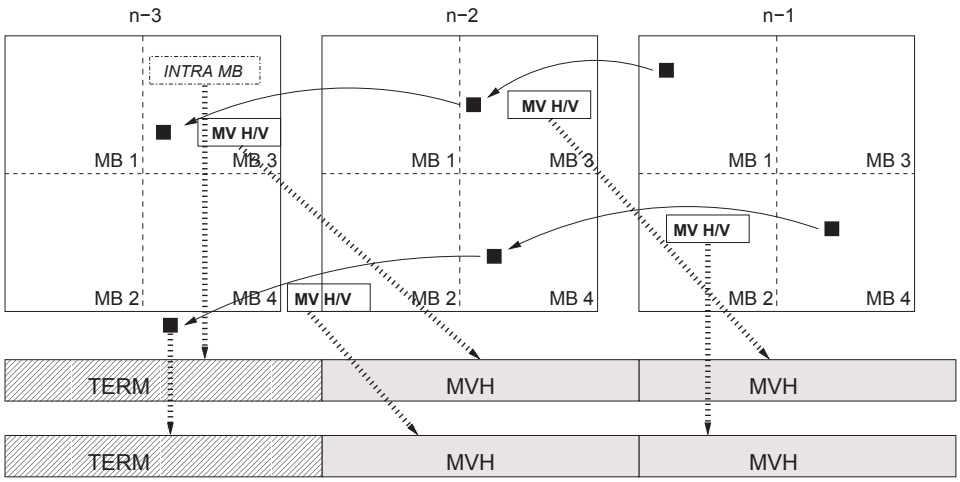



**Figure 3.12** – Generation of the MV history [135].

$$FMV_{i,j}^{n-1}d = iF - i, \;\; d = h \tag{3.23}$$

$$FMV_{i,j}^{n-1}d = jF - j, \;\; d = v \tag{3.24}$$

•**Step 3:** The third step consists in a regularization to fill possible discontinuities in $FMV$, which is implemented by applying a two dimension median filter based on a $12 \times 12$ window [136]. Edge preserving of the motion fields was also taken into consideration to avoid loosing important details.

•**Step 4:** Reconstruction of the missing frame $n$ is implemented in the fourth step. Since the $FMV$ motion field has half pixel resolution, each pixel of a reconstructed frame is represented by a $2 \times 2$ mask, increasing the resolution by a factor of two. It is also taken into consideration that more than one MV $FMV$ will lead to the same pixel of the reconstructed frame. In such cases, a weighted average of the multiple contributions is performed.

•**Step 5:** In the fifth task, the missing pixels of a reconstructed frame are interpolated. Just like in the previous step, some MVs can point to the same pixel and there are

also some pixels that do not have any MV that point to them. To fill these void areas, the median of the pixels around the missing one is computed using a $7 \times 7$ window.

•**Step 6:** The final step is to downsample the reconstructed frame with half-pixel resolution to the original frame size. This is performed by averaging each $2 \times 2$ block into single pixel values, resulting in a reconstructed frame with the target size.

The authors reported good results and significant PSNR gains over the EC reference method used for comparison (TR). No random losses were considered in the performance evaluation, the tests were only made at fixed frame numbers, starting from the fourth frame. This way, there are always four uncorrupted frames available at the beginning of the GOP sequence, which is of major importance for the concealment performance of this method. The gains over the TR in individual frames can be up to 8dB, but no study is performed on the impact of the recovered frames over the video sequence, i.e., only local losses are considered. In high motion sequences, higher PSNR gains over TR are obtained. However, the method should be optimized for low motion sequences, where the concealment should be easier, but in this case TR achieves slightly better results than the proposed method (i.e., up to 0.55dB). The authors only considered the decoding process until the occurrence of a frame loss. In such cases, the lost frame is concealed and decoding does not continue, which does not allow to evaluate the impact of error propagation. The proposed method is based on a very interesting idea, which considers not only the motion information on the nearest correctly decoded frames, but also several frames are used in the error concealment process in order to achieve more accurate reconstructed frames. However its performance was not evaluated in realistic scenarios, which may limit the reported advantages in more generic application scenarios.

Yu Chen in [137] proposed an EC method for full frame loss in here, the main difference is that MVE extrapolation is performed at the pixel level, rather than block level. Additionally, bi-directional error concealment uses both the previous and future frames. In Figure 3.13 there is an example of a typical block-based MVE, using four MBs ($MB_1$, $MN_2$, $MB_3$ and $MB_4$) to extrapolate the MV for the lost region. Such MV corresponds to the largest overlapped area, which in this example is the MV associated to $MB_2$. As shown in Figure 3.13, despite of $MB_2$ being the biggest overlapped region, there is still a large area that is not overlapped. Therefore, the usage of the same MV to conceal the

whole lost MB could result in high number of block artefacts in the recovered region. To minimize this problem, Yu Chen proposed an MVE technique operating at the pixel level. When there exists multiple MVs, due to multiple overlapped regions (marked as "$\circ$"), all available MVs are used to extrapolate the lost pixel. In cases where there are no overlapped blocks, such as the regions marked by a "$\triangle$", the MV from the co-located pixels of the previous frame is used, similar to motion copy method (MoCp).

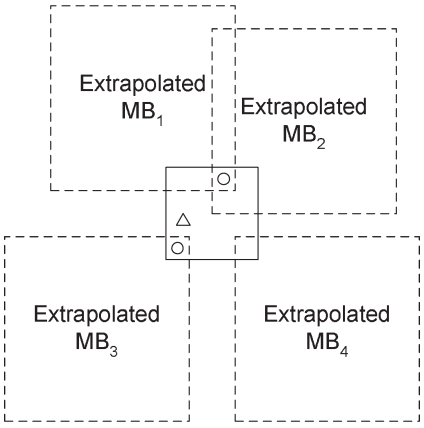



**Figure 3.13** – Pixel based Motion Vector Extrapolation example (pMVE) [137].

Considering the lost frame $n$, bi-directional interpolation performed by applying the pixel level MVE (pMVE), described above, in both frames $n-1$ and $n+1$. The bi-directional interpolation of a determined lost pixel $p(x, y)$ belonging to frame $n$ is defined by Equation 3.25. $x$ and $y$ are the respective coordinates, $pf(x, y)$ and $pb(x, y)$ are, respectively, the pixels recovered by using MVE on $n-1$ and $n+1$. By choosing $w = 0.5$, Equation 3.25 defines an arithmetic average between $pf(x, y)$ and $pb(x, y)$.

$$p(x, y) = w \times p_f(x, y) + (1 - w) \times p_b(x, y) \tag{3.25}$$

The authors compared the proposed bi-directional EC method with MVE [134], MMA [135] and motion compensation. In motion compensation all MVs of the corrupted frame are correctly received, therefore only residual information is lost. Thus, it is possible to evaluate and compare more accurately the reliability of the interpolated MVs that were computed by the bi-directional method, MVE and MMA methods. The proposed method is able to achieve good results, losing in average 0.50dB over the optimal motion compensation, but achieving PSNR gains of 0.93dB over MVE and 1.06dB over MMA.

B. Yan proposed in [138] an hybrid frame error concealment algorithm (HMVE) for H.264/AVC, based on the block based MVE method [134] and the pixel based MVE (pMVE) [137]. The author claims that, despite the improved performance of pMVE over MVE, for pixels without available MVs from the extrapolated blocks ("△" in Figure 3.13), MVs from the co-located regions of the previous frame are used. However, in many situations they are not accurate enough, so it is very likely that pixels belonging to regions defined by "△" are incorrectly interpolated. This mainly occurs in sequences with high motion.

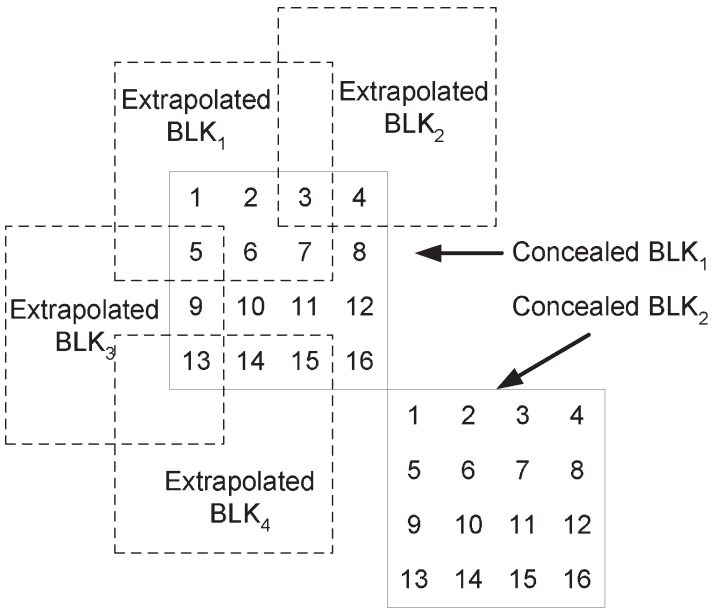



**Figure 3.14** – Example of Hybrid Motion Vector Extrapolation (HMVE) [138].

In HMVE, the pixel classification is different from that of pMVE. Figure 3.14 shows an example of the pixel classification for the concealment of two lost blocks (recovered $BLK_1$ and recovered $BLK_2$) using four available extrapolated blocks (Extrapolated $BLK_1$ to Extrapolated $BLK_4$). Pixel classification is performed by categorising pixels into three parts:

**Part A:** Pixels that belong to *Part A* are covered by at least one extrapolated block. In Figure 3.14, those pixels belong to *Concealed* $BLK_1$ and are numbered as {1,2,3,4,5,9,13,15}.

**Part B:** Pixels belonging to *Part B* are those not overlapped by extrapolated blocks, but belong to a recovered block that has some overlapped pixels. In the example of Figure 3.14, pixels in this situation belong to *Recovered* $BLK_1$, being numbered as {8,10,11,12,16}.

**Part C:** Pixels belonging to *Part C* are the ones not having any overlapped extrapolated blocks and belong to a recovered block without any overlapped regions with extrapolated blocks. In Figure 3.14, pixels meeting these conditions belong to *Recovered* $BLK_2$ and are numbered [1,16]. Since HMVE was designed for H.264/AVC, the smallest block size for motion compensation is $4 \times 4$, the concealment process is also based on this block size, as shown in Figure 3.14. In HMVE, using the pixel classification described above, a different approach is used to interpolate the lost MVs, when comparing to the original MVE and pMVE. First, two MV are extrapolated based on MVE pMVE and, then a selection between both MV candidates is performed.

Considering that $B_n^i$ is a $4 \times 4$ lost block to be recovered, the two MV candidates for $B_n^i$ are represented by $MV_m(B_n^i)$ and $MV_a(B_n^i)$, $m$ denotes MVs corresponding to the extrapolated MV with the maximum similarity with $B_n^i$, while $a$ denotes the MV computed as a weighted average of all extrapolated MVs of blocks that are overlapped with $B_n^i$.

The blocks extrapolated from the reference frame to corrupted frame $n$ being concealed are defined as $EB_n^j$, where $j$ is the number of the extrapolated blocks. The weight $w_n^{i,j}$ for each extrapolated MV is obtained by computing the number of the corresponding overlapped pixels of $EB_n^j$ on top of $B_n^i$. The higher is the weight $w_n^{i,j}$, the higher will be the correlation between $EB_n^j$ and $B_n^i$. $MV_a(B_n^i)$ is then defined by Equation 3.26:

$$MV_m(B_n^i) = MV(EB_n^{j^*}), \;\; j^* = arg\ max\{w_n^{i,j}\} \tag{3.26}$$

The average MV of the overlapped extrapolated blocks is defined by $MV_a(B_n^i)$ in Equation 3.27

$$MV_a(B_n^i) = \frac{\sum_{j=1}^{M} MV(EB_n^j) w_n^{i,j}}{\sum_{j=1}^{M} w_n^{i,j}} \tag{3.27}$$

If $B_n^i$ is not overlapped by any extrapolated block $EB_n^j$, then the recovered MV will be set

to zero. After the previous MV extrapolation, a set of extrapolated MVs is obtained and defined as $MVS_p(P_n^{x,y})$ for each pixel $P_n^{x,y}$ located at $(x, y)$ and belonging to block $B_n^i$. Then, the pixels for each of the three parts, a new set of MVs ($MVS(P_n^{x,y})$) is computed as follows:

**Part A:** For each pixel belonging to Part A, the corresponding set of MVs $MVS(P_n^{x,y})$ is defined by Equation 3.28:

$$MVS(P_n^{x,y}) = \{MV_m(B_n^i), MV_a(B_n^i), MVS_p(P_n^{x,y})\} \tag{3.28}$$

Some MVs belonging to MVS, might be wrongly extrapolated. In order to exclude such wrong MVs, the differences between each MV are computed. Only those MVs that have the smallest distances between them, i.e., smaller than a predefined threshold $T$, are considered to recover $P_n^{x,y}$.

**Part B:** MVs for Part B are computed as defined by Equation 3.29:

$$MVS(P_n^{x,y}) = \{MV_m(B_n^i), MV_a(B_n^i)\} \tag{3.29}$$

**Part C:** MVs for Part C are computed as defined by Equation 3.30. For the pixels belonging to this part, the MVs are obtained by using the co-located ones from previous frame $n - 1$.

$$MVS(P_n^{x,y}) = \{MV(P_{n-1}^{x,y})\} \tag{3.30}$$

After computing the MVs that correspond to each part, the final MV for a certain $P_n^{x,y}$ is obtained by computing the average of the available MVs stored in $MVS(P_n^{x,y})$.

The authors have shown that HMVE is able to achieve good results when compared to pixel based MVE (pMVE). HMVE is able to significantly improve the performance of pMVE, mainly due to the capability of HMVE to obtain more accurate MV, especially in regions without extrapolated blocks, as those defined as *Part B* and *Part C*. HMVE is able to achieve a PSNR gain up to 1.15dB in the tested sequences, considering only the corrupted frames.

H. Liu, in [139] uses a bi-directional method that exploits motion information from previous and future frames. The author assumes that the corrupted frame $n$ is only detected when the decoding process of $n+1$ starts. At this point, the MVs of $n+1$ are available.

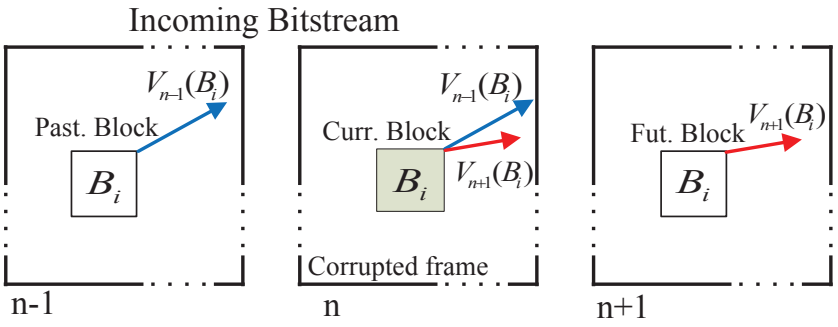



**Figure 3.15** – Bi-directional motion copy EC [139].

Figure 3.15 shows an example of how a future $(n+1)$ and past $(n-1)$ MV is used in bi-directional MoCp. $B_i$ is the current block being recovered, while $Vn(B_i)$ is the MV of the corresponding block $B_i$. Considering that frame $n$ of the incoming bitstream is corrupted, as shown in Figure 3.15, to recover the lost block $B_i$ motion compensation is used to interpolate the lost pixels $p_t(x, y)$.

The recovered pixels at coordinates $(x, y)$ for the lost block $B_i$ are interpolated following Equation 3.31 $(\hat{P}_t(x, y))$, which defines the average of the motion compensated pixels using the backward and forward MVs. Motion compensated pixels with backward MVs $(\hat{p}_t^B(x, y))$ are defined by Equation 3.32 and pixels compensated with forward MVs are defined by Equation 3.33.

$$\hat{P}_t(x, y) = \frac{\hat{p}_t^F(x, y) + \hat{p}_t^B(x, y)}{2} \tag{3.31}$$

$$\hat{p}_t^B(x, y) = P_{t-1}(x + V_{t+1}^x(B_i), y + V_{t+1}^y(B_i)) \tag{3.32}$$

$$\hat{p}_t^F(x, y) = P_{t-1}(x + V_{t-1}^x(B_i), y + V_{t-1}^y(B_i)) \tag{3.33}$$

This method shows better performance than the original MoCp technique. By using past and future MVs, the recovered MV has a better chance to represent more precisely the

original motion. The authors claim that bi-directional motion copy method achieves better performance, resulting in increased PSNR gain over the reference methods [123, 140] and MoCp up to 0.45dB, 0.42dB and 2.21dB, respectively.

## 3.4 Error concealment for 3D and multiview-video

The existence of different formats for 3D visual representation requires different approaches and methods to deal with data loss. As mentioned before, the most popular 3D formats are the MVC and MVD. Many EC methods presented in the literature for MVC and MVD are based on the 2D techniques described in the previous sections of this chapter. Depending on the type of losses, spatial and temporal information are used, but in these multiview formats, new types of information can be exploited in order to improve the concealment performance. In the case of MVC, besides spatial and inter-frame correlations, inter-view correlations can also be exploited. The similarities between several frames, at a certain instant of time, from different viewpoints is high, thus such type of similarity may be exploited. In the case of MVD, besides having different viewpoints as verified in the case of MVC, for each texture frame there is also a corresponding depth map. As mentioned before, texture and depth information can be used together in order to synthesise new virtual viewpoints. Using the same principle as in some coding methods, the correlation between depth maps and texture is also used in order to improve concealment performance. In this section, a review of the most relevant EC methods for MVC and MVD is presented.

S. Liu in [141] proposed a full frame EC method for MVC. This approach takes into consideration the same principles of motion similarities between frames in the temporal domain, as in the case of 2D video EC. Nevertheless, in this case the redundancy between adjacent views is used. As the EC method for 2D video described in [135], a motion field is computed based on the previously decoded frames. S. Liu proposed a similar approach, but taking advantage of the MVC video characteristics, where the motion information of the adjacent views is used to recover the corrupted frames. The motion field of the adjacent views is based on a simplified global disparity model, where the MVs from adjacent views are considered to be similar, simply taking into account the camera displacement. When a frame from a non-base view is lost, each lost MB is recovered

by copying all the block partitions and respective MVs from the adjacent view. When motion information is not available in the adjacent view, as in intra-coded MBs, spatial error concealment is performed by setting the lost MB being concealed as a skip in P-coded pictures or direct mode, in the case of B-coded pictures [115]. But, S. Liu compared the proposed method only with temporal replacement method (TR), where gains up 2.6dB and 0.97dB, on average, were achieved for the tested sequences. Despite these good results, this method still has room for improvement, mainly due to simplistic EC approach that uses the same global disparity for all concealed MBs. This might be an advantage in terms of computational complexity, but may reduce the EC accuracy when compared with the case where individual disparity values would be used for each recovered MB.

Y. Chen in [142] also proposed also a full-frame error concealment for stereoscopic video using a frame difference projection based on disparity (DFDP) of a stereo pair. Figure 3.16 shows an example of how the temporal similarity between frames (inter-frame) and also between views (inter-view) is exploited. The method assumes that the left view is the base view, meaning that this bitstream is independent and can be decoded without the need of other views. In the case of the right view, correlation between views was taken into account, meaning that the adjacent views are needed in the decoding process. Y. Chen proposed a method that is targeted for non-base encoded views, where it is also exploited inter-view information.

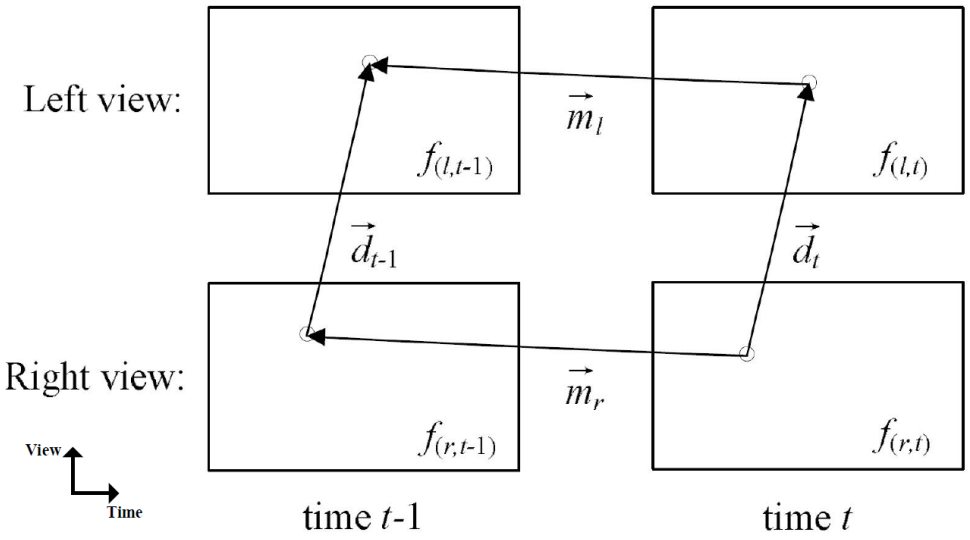



**Figure 3.16** – Inter-view and inter-frame correlations [142].

The left view is defined by $l$, the right view by $r$, and $t$ defines the time instant. A pixel from $f(r,t)$ of a corrupted frame can be obtained by motion and disparity vectors, one due to motion activity ($\vec{m_l}$ and $\vec{m_r}$) and the other due disparity between frames ($\vec{d_t}$ and $\vec{d_{t-1}}$). Assuming that objects in scene do not change significantly along time, nor between frames from different views, it is reasonable to consider that the MVs $\vec{m_l} \approx \vec{m_r}$ and $\vec{d_t} \approx \vec{d_{t-1}}$, indicating that both inter-view and inter-frame correlations are high. Y. Chen DFDP method is based on this assumption and comprise three main functions: 1) Change Detection, 2) Disparity estimation and 3) Frame difference projection.

In the first function, Change Detection, a temporal change detection is performed by computing the absolute frame differences for all pixels between a certain temporal instant $t$ and $t-1$ in the left view $l$. The resulting matrix $\Delta f$, defined by Equation 3.34, represents the corresponding frame difference, which is then filtered with a mean filter, followed by a thresholding [143]. In order to detect the moving objects, pixels belonging to the foreground and background are separately identified. This filtered matrix is represented by $M_{(l,t-1\longrightarrow t)}(x,y)$ and defined by Equation 3.35, where $x$ and $y$ are the corresponding pixel coordinates. The threshold $T$ is computed by an iterative algorithm, as described in [142].

$$\Delta f_{(l,t-1\longrightarrow t)}(x,y) = |f_{(l,t)}(x,y) - f_{(l,t-1)}(x,y)| \tag{3.34}$$

$$M_{(l,t-1\longrightarrow t)}(x,y) = \begin{cases} 1, & \Delta f_{(l,t-1\longrightarrow t)} \geq T \\ 0, & \text{otherwise} \end{cases} \tag{3.35}$$

In the second function, the horizontal disparity estimation is computed between the stereo pair $l$ and $r$ (a parallel camera arrangement is used). The authors considered that a disparity vector could be decomposed into two components, the global and the local disparity. The global disparity $d_{global}^{(t-1)}$ is computed for the regions where temporal changes occur, as defined by Equations 3.36 and 3.37. The objective is to compute the disparity $d_{global}^{(t-1)}$ where the absolute differences between frames belonging to left ($l$) and right($r$) views are smaller.

$$E_{global}^{t-1} = \sum_{M_{(r,t-1\longrightarrow t)}(x,y)=1} |f_{(l,t-1)}(x,y) - f_{(r,t-1)}(x-d,y)| \tag{3.36}$$

$$d_{global}^{(t-1)} = arg\ \min_{d}\ E_{global}^{t-1} \tag{3.37}$$

The local disparity $d_{local}^{(t-1)}$ is computed using an $m \times n$ window, as defined by Equations 3.38 and 3.39. To compute the local disparity, an $8 \times 8$ window was used and a search range defined by $d \in [-20, 20]$.

$$d_{local}^{(t-1)} = arg\ \min_{d}\ E_{local}^{(t-1)}(x, y, d) \tag{3.38}$$

$$E_{local}^{t-1}(x, y, d) = \sum_{\gamma=-\frac{m}{2}}^{\frac{m}{2}} \sum_{\xi=-\frac{n}{2}}^{\frac{n}{2}} |f_{(l,t-1)(x+\gamma,y+\xi)} - f_{(r,t-1)(x+\gamma-d-d_{global}^{t-1},y+\xi)}| \tag{3.39}$$

After obtaining the global and local disparity components, the final disparity values $d^{t-1}(x, y)$ are expressed as:

$$d^{t-1}(x, y) = d_{global}^{t-1} + d_{local}^{t-1}(x, y) \tag{3.40}$$

In the third and final function of this EC method, frame difference projection is performed based on the inter-frame and inter-view correlation, knowing that MVs $\vec{m_l} \approx \vec{m_r}$ and $\vec{d_t} \approx \vec{d_{t-1}}$. The change detection map $M_{(r,t-1\longrightarrow t)}(x, y)$ and the temporal frame difference $\Delta f_{(r,t-1\longrightarrow t)}(x, y)$ of the right view can be computed based on the left view $M_{(l,t-1\longrightarrow t)}(x, y)$ and $\Delta f_{(l,t-1\longrightarrow t)}(x, y)$. It is considered that the lost frame is the right view from temporal instant $t$ ($f_{(r,t)}$), which is recovered using pixels from $f_{(r,t-1)}$ together with the temporal distances $\Delta f_{(r,t-1\longrightarrow t)}(x, y)$, as defined by Equation 3.41:

$$f_{(r,t)}(x, y) = f_{(r,t-1)}(x, y) + \Delta f_{(r,t-1\longrightarrow t)}(x, y) \tag{3.41}$$

To validate the method, Y. Chen compared with two other techniques [144, 145], which also exploit the correlation between views to recover the lost regions. The proposed method is able to achieve good results and to surpass the best reference methods by an average Y-PSNR of 1.42dB. It would be also interesting to compare the proposed method with other popular EC methods tailored for 2D video, such as temporal replacement (TR)

or motion vector extrapolation (MVE), in order to conclude more clearly the accuracy of the proposed method over other techniques.

T. Chung, proposed an EC in [146], which is similar to the previous one [142]. This approach also exploits the correlation between different views, in order to extract MVs from uncorrupted views to the one being concealed. The novel idea of this work is to consider the occlusions between views [142]. Figure 3.17 shows how the occlusions between the views are detected. For example, a scene composed by points *a, b, c, d, e, f, g*, which are represented in two distinct frames at the same temporal instant from two views, e.g., a stereo pair. Considering an object that is sitting at a certain distance from the background represented by pixels *a* and *b*, some pixels of the background are not visible by both views. These pixels belong to the occluded region, represented by *c, d, e, f*. The EC cannot be accurate for these pixels, as presented in [142], therefore the authors propose a technique to fill this region. After recovering the lost region corresponding

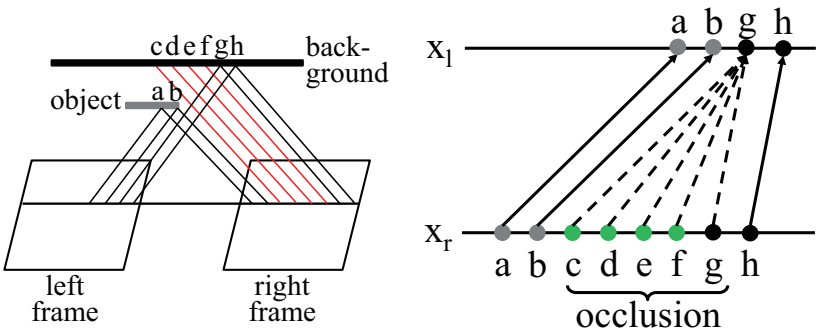



**Figure 3.17** – Detection of occluded region between views [146].

to the non-occluded areas, the hole filling is performed not only on the occluded areas, but also in other regions that temporal EC was not successfully performed. In some cases, some MVs can point to the same pixels, resulting in empty regions that were not concealed. The non-concealed regions are filled, by checking the motion activity of neighbours previously recovered. Using an $5 \times 5$ window around the empty pixel, and if the motion intensity from such window is below a pre-defined threshold, temporal replacement (TR) is used. When the motion activity is above the threshold, the holes are filled with spatially nearest available pixels. Comparing to the implementation in [142], on which the method of T. Chung is based, the authors reported an average PSNR gain of 0.4dB. In the case of individual frames, the PSNR gain over the method in [142] can be over 1.7dB,

which is quite a significant improvement. Since the average PSNR advantage over the best reference methods is not very high (0.4dB), the proposed method would be better validated if subjective tests were made, in order to verify if such a small PSNR difference has some effect on the perceived quality by the viewers.

B. Micallef in [147] and [148], presented an MVC EC algorithm based on the FMO H.264/AVC resilience tool. Since the adopted EC techniques used in this work cannot deal with full-frame loss, the authors rely on the assumption that when errors occur, only portions of the images are lost (e.g. slices). As the previous methods described in this section [142, 144, 145], this method also exploits both inter-view and inter-frame correlations but, in this case, the similarities are not exploited by computing a disparity for the corrupted pixels. B. Micallef used a more specific approach, less computationally expensive, because the used disparity MVs (DVs) are extracted from the MVC bitstream. In MVC, to increase coding efficiency inter-view correlations are also exploited in non-base views. Based on this DVs, the authors adopted EC techniques that were primarily developed for 2D video, more specifically the one published in [125] and also described in Section 3.3.3. Authors used the same method as in [125] but also adding the DVs as candidates. The MV or DV producing the smaller distortion at the boundaries of the lost region is chosen to recover the corrupted MB. Depending on the MVC coding structure being used, DVs might not be available for concealment. In case of the first ($view0$), which is typically the base view, DVs are not available and the EC is performed in the same manner as 2D video. In the other views, anchor frames only have available DVs and not MVs because in such frames only inter-view prediction is exploited in the coding process. In the other frames, MVs and DVs are available in EC. As mentioned before, B. Micallef tested the proposed EC methods for MVC using FMO, and also using another coding scheme that uses a fixed slice size of 150 bytes. Using FMO or the fixed slice size, the authors reported that the accuracy is able to surpass clearly the scheme where a fixed slice size is used. Comparing a reference method (i.e. FC) with the proposed method, PSNR gains over 2dB are achieved, proving its effectiveness. Nevertheless, more EC references could be used, in order to evaluate more clearly the gains obtained from the use of disparity information (DVs).

Stankiewicz proposed in [149] an EC algorithm for multiview video, but in this case, the MVD format was used. A novel approach was employed in order to conceal corrupted texture frames based on DIBR. The major novelty of this work is to use of DIBR in EC,

where the synthesis of a virtual view is used to recover the lost areas. Besides DIBR, also intra-based and temporal techniques are used. In the DIBR process, the author considered that the depth maps are available at the decoder by either being transmitted through different channels or generated on-site. First, the lost regions are recovered using a combination of the inter-view (DIBR) and temporal techniques. Since, the inter-view and temporal technique might not be able to fill all pixels of the lost regions, the remaining areas are recovered using an intra technique, which simply uses the average of the nearest available pixels. Regarding, the inter-view and temporal techniques, these two are used to recover all the missing regions. After performing this task, only one of the methods is chosen based the estimated accuracy of the EC technique. Although using demanding simulation conditions with high PLR (50%), the authors considered a very specific loss scenario where the corrupted frames are not used as references; therefore error propagation does not exist. The tested scenario should be more realistic and consider other types of losses to verify the EC method ability to mitigate the negative effects of error propagation like other typical EC algorithms. Stankiewicz assessed the quality of the proposed EC method by comparing it with two other reference methods, temporal replacement (TR) an another temporal EC method, similar to the one described in [139] and in the previous Section 3.3.4. The quality was measured through a set of subjective tests and it was found that the proposed EC method achieves better results than the reference methods for almost all sequences. Only for one sequence the proposed method did not achieve better results, which was justified by the inconsistency in the lightning environment between cameras, which affects severely the accuracy of inter-view concealment.

Another EC algorithm was proposed by X. Xiang [150] for stereoscopic video, which is based on an autoregressive model (AR) [151]. This method starts by acquiring the motion information (MVs) and also the disparity vectors (DVs) of the corrupted regions. Then, the AR coefficients are computed based on the spatial correlations using both the previously acquired MVs and DVs. The final step is to apply the AR model on all pixels of the lost regions, using weighted interpolation of the selected prediction directions. Note that each of the MVs and DVs are refined using BMA, then the best MV and DV is chosen for the recovery process. Each pixel of the lost region is computed by using a weighted interpolation of the pixels that belong to a window with size $(2 \times R)$. $R$ is the radius of this window, centred on the pixel located at the point given by MV or DV from the reference frame(temporal-correlated or interview frame). Figure 3.18 shows an example

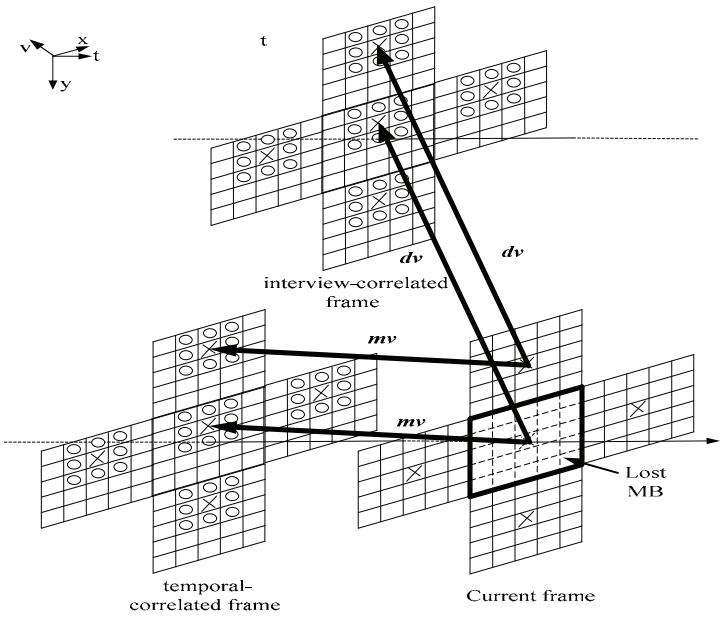



**Figure 3.18** – Temporal and interview EC model [150].

of pixels selection to compute the weighted interpolation. A cross ($\times$) in the lost MB of the current frame defines the lost pixel, while a circle ($\bigcirc$) defines the surrounding pixels in the reference frames that are contained inside the region defined by $2R$. Pixels defined by $\bigcirc$ are used in the weighted interpolation and each of them has an associated weight $\alpha$ that is computed by the proposed AR model. The authors reported PSNR gains up to 1.28dB over the error concealment method implemented in JM H.264/AVC reference decoder, for the base view (without inter-view redundancy to exploit). For the second view, where inter-view redundancy exists, the PSNR gain over JM is higher, up to 3.32dB, revealing the importance of using DVs in order to achieve an accurate EC.

A. Ali in [152] present a spatial EC method for MVD, which uses the depth information to restore the corresponding texture image, based on simple thresholding of the depth map. This approach shows high potential, since the main edges of the depth map define the limit between the background and foreground, therefore can be used to conceal the corrupted texture. The validation of the proposed method is weak because a simple error pattern was used. Only one square block ($16 \times 16$) or three consecutive square blocks were simulated as corrupted regions. Using such limited error patterns and small error percentages, it is difficult to evaluate the accuracy and to validate the performance as a general result. Moreover, since the 3D format is MVD, the quality assessment of the

synthesised images using the recovered texture images should have been performed, in order to evaluate the impact of the recovered texture images by DIBR.

In the work described in [153], B. Yan proposed an EC technique for MVD which has many similarities with his previous work regarding EC for 2D video [138] (described in Section 3.3.4). The main focus of this paper is to conceal corrupted texture, but taking advantage of additional information given by the depth maps. HMVE and PMVE EC techniques are implemented with the addition of depth support, in order to exploit the advantage of using depth maps in the error concealment process. First, a set of extrapolated MVs for each pixel of the lost region is computed using HMVE and PMVE. Then, the depth value of each MVs point is checked based on the assumption that objects present in the scene with similar motion have also similar depth values. When depth maps are lost, no specific EC is performed for depth maps and the traditional PMVE or HMVE for 2D video is used. Since some MVs might point to the same pixels, some other lost pixels might not be recovered. In such cases, a simple weighted interpolation of the nearest available pixels that were already recovered using the previous techniques. The authors reported that the use of depth map information to recover corrupted texture in both PMVE and HMVE EC algorithms is a significant advantage. Without the use of depth, the HMVE presents a better PSNR than PMVE. With the addition of depth maps, the advantage of the HMVE method is even more significant. As other EC techniques from other authors described in this thesis, the evaluation of the synthesised images is not performed. Therefore, it is also not possible to evaluate the effect the recovered depth and texture on the synthesis of virtual views.

V. Doan proposed in [154] a view synthesis based error concealment technique, where the main goal is to recover corrupted texture images. In the case of a corrupted depth map, the error concealment technique is described in [155]. For the sake of simplicity, the author considered an application where two views with the corresponding texture and depth maps are used. The left view is considered to be error free. This approach is also based on the assumption that slices containing eighty macroblocks are randomly lost. Regarding the EC of the texture, which is the main focus of this method, when a missing region of the right view is detected, the corresponding pixels of such regions are synthesised through DIBR, in order to find the matching pixels of the left view. In the synthesis process, some pixels may not be synthesised if these pixels belong to occluded regions. If the number of unsuccessful synthesised pixels is higher than a determined

threshold, the corrupted region is recovered using conventional methods [155]. Otherwise, the following steps are performed in order to choose a temporal or synthesis based error concealment method:

Firstly, an inter-view MV prediction is performed by computing the DVs, using the available depth of the corresponding corrupted texture image. Since such DVs are block-based, they may be somehow inaccurate at the pixel level. To overcome this problem, the author adopted the block partition scheme of H.264/AVC, using the disparity motion field computed from the depth maps. It allows MB partitions from $16 \times 16$ pixels down to $8 \times 8$ pixels, and the partition size is chosen according to the one that produces less deviation from the disparity MV field.

Secondly, view synthesis based selection is performed by using not only the typical temporal MVs (zero-MV, neighbour-MVs), but also the DVs computed in the previous step and the co-located depth MVs. The best MV/DV is selected by computing the distortion between the predicted block and the corresponding synthesised texture. Besides considering the distortion between predicted block and synthesised texture, the distortion at the boundary of the recovered region is also taken into account in order to achieve a more accurate spatial smoothness.

Finally, a selection is performed between a Temporal Error Concealment (TEC), using the MVs/DVs of the previous steps, or simply by synthesising the missing regions. This is performed by selecting the technique that produces less distortion at the boundaries of the missing region, computed using BMA.

V. Doan compared the proposed method with BMA, DMVE and also the technique described in [155]. Compared to the best reference method [155], the results show that the proposed method is able to achieve improvements up to 2.19dB, at 20% PLR. To validate these results, and to allow comparison with other published works, the quality of the synthesised views using the recovered texture and depth maps should have been evaluated.

## 3.5 Depth map error concealment for MVD

Most of the work developed so far in the field of depth map error concealment is based on the use of temporal information extracted from video-plus-depth decoded streams. The most common data loss scenarios involve the loss of a full frame, but depending on the coding/packetisation method, also single blocks or groups of blocks are lost (bursts). As mentioned before, despite the depth maps are not used for display, they play a very important role in the overall quality of the synthesised views, affecting significantly the synthesis quality. Therefore, in error-prone environment it is crucial to use EC methods that can mitigate the effects of errors in depth maps that would lead to inaccurate synthesis. In this section, a review of the most important EC methods for depth maps is presented, having in mind that some of these publications do not refer exclusively to depth map EC, but also to recover corrupted texture. Depth map EC algorithms are strongly influenced by the existing techniques for 2D and 3D video that were previously described in this chapter. As in 2D and 3D, some techniques rely only the available spatial information to recover the lost regions, while other techniques rely on temporal and inter-view information. Finally, other techniques rely on the combination of all approaches.

The research work described in [156] exploits the redundancy in motion information between texture images and their corresponding depth maps to recover lost regions, in both the texture images and depth maps. However, despite the good objective results, the depth contours are not preserved and some blocking artefacts are noticeable in the depth map, mainly at the edges between the foreground and the background. As pointed out before, this leads to poor quality in the synthesised views. In [155] the authors assume that in presence of depth map errors, the corresponding texture image region is free of errors, then, it can be used for the depth map recovery. Although the authors report good objective results, a limitation of this work lies on the fact that the impact on the quality of the synthesised images of other views is not evaluated. In another recent work [157], the authors propose an EC method for both texture and depth, but additionally based on temporal information. In many of the referenced publications, the impact of the error concealment accuracy in texture and depth maps in synthesis process is not evaluated, which is rather important since one of the main advantages of using MVD is the capability of synthesising other virtual view points.

B. Yan proposed in [158] an EC method for 3D video transmission, based on H.264/AVC, that is used on both texture images and depth maps. In this work, the authors use a BMA technique together with the available motion information to recover both corrupted depth maps and texture images. As in the other EC techniques with a similar approach, the BMA technique is used to select the best MV candidate, but in this case, the MV candidates are not only from the uncorrupted spatially and temporal neighbouring regions, but also from the corresponding depth map. The use of motion information from depth maps to texture and vice-versa can be questionable. To exploit redundant motion information between texture and depth, the MV sharing process should not be done indiscriminately, because not all MVs of the texture are suitable to be used for the depth, and vice-versa. This issue is much more evident when using depth MVs in texture, because these MVs are much more likely be too inaccurate, while the texture MVs are much more correlated with depth maps. This is mainly due to the fact that many objects in the scene might have motion, but at the same depth level and in this type of region the depth does not change significantly. Typically, the depth MVs that are more similar to those of the texture are those located at the objects edges. At the edges of an object, sharp changes in depth maps values occur, and then MVs at such locations are more likely to have a high motion correlation with texture images motion. As mentioned before, B. Yan used depth MVs as additional candidates in the BMA technique to recover texture errors. It would be clearer if it is known how frequently the depth MVs are used, when compared to texture MVs of neighbouring regions. Despite the good results of the EC accuracy in the texture video, the authors did not evaluate the quality of the reconstructed depth maps. Furthermore, in these experiments only one video sequence was used. Since the proposed method is intended to be used with the MVD format, the quality of the synthesised views using the recovered texture and depth should also be evaluated. Therefore, it is very difficult to get a precise idea of the accuracy of the proposed method for synthesised images.

C. Hewage in [156, 159] proposed a whole frame EC for depth maps in MVD format based on the SVC coding architecture, where the depth maps are encoded as an enhancement layer. In this EC method, to recover lost texture or depth frames, a similar approach as previously described by B. Yan in [158] is used. In this method MVs from the corresponding depth maps or texture are used to conceal the error effects in the respective corrupted depth map or texture. Since in the C. Hewage approach, the EC method is intended to be used for entire frame loss, BMA is not used to refine the MVs, as in the

method proposed in [158]. In this case depth or texture MVs are used directly in the error concealment process. To achieve an improved depth EC performance, MVs from texture images are also used in the enhancement layer. Besides the obvious increase of bitrate (BR) of this solution, the author limited the depth BR up to 25% of the corresponding texture BR. In the EC method assessment, C. Hewage did not evaluate the quality of synthesised images, but only the individual PSNR of both texture and depth. PSNR is a widely accepted metric for 2D, but not the most adequate to evaluate 3D video depth maps. The PSNR of a depth map might be higher than another one and the synthesised image using such depth map might be worse. That is why it is important to evaluate error concealment performance of depth maps by analysing the synthesised images computed from the recovered data. Still, by looking at the available results, it seems that sharing motion information from the texture to depth maps and vice-versa seems to be quite effective because very good PSNR results are achieved. C. Hewage in [156] presented an example where the proposed method is able to achieve good results when compared with TR methods. However, since the depth maps are not used directly for display, subjective tests and/or assessment of the synthesised image should have been done to confirm it.

Y. Liu in [155], presented a TEC method for texture and depth maps, which is also based on the correlation between these two. This method is more focused on recovering texture images, using additional MV candidates provided by the depth maps. Regarding EC of depth maps, the authors used a similar approach as described in [156], where the MVs from the texture are directly used to recover the lost regions in the depth maps. Y. Liu claims that depth does not change dramatically through time, and then using MVs from texture is an effective approach. This claim is not totally correct since, in very fast moving areas, the texture MBs are usually coded as intra or the corresponding MVs in this type of fast moving regions are not very accurate. In such fast moving areas, the texture MVs are not accurate enough to perform the EC in the depth maps. The authors did not take these facts into consideration, and besides the good results presented for the EC of texture, when compared with BMA and DMVE, the quality of the depth maps are simply evaluated by using the PSNR metric. Quality assessment of the synthesised views was not performed.

T. Chang in [160] proposed a temporal EC method for stereoscopic video, for both texture image and depth maps using MVD. Regarding the texture, MVs for the lost frames are obtained by performing an extrapolation of the temporally neighbouring and uncorrupted

frames. Depth maps errors are concealed by exploiting the correlations between texture and depth of the same view and from the adjacent view, depending whether the lost frame belongs to the base view or to the second view. The prediction structure is used, as shown in Figure 3.19, assuming that the base view is the left one and the non-base view is the right one. The GOP structure has a length of eight pictures ($T_i,\ i = 0, 1, ..., 7$), while $C_l$ and $C_r$ defines the respective left and right texture frames. $D_l$ and $D_r$ defines, the left and right depth maps, respectively.

In the encoding process, besides temporal correlation (motion compensation prediction, MCP) also the redundancy between views is exploited (disparity compensation prediction, DCP). In the base view, only MCP is used and in the non-base view MCP and DCP are used. Since texture and depth maps are separately encoded, MCP and DCP are only used within the same type of data, as shown by Figure 3.19. In this approach, the correlations between depth and texture are not exploited in the coding process. Depending whether the lost frame belongs to the base or non-base view or the availability of the corresponding texture and depth frame, the error concealment approach might differ. The EC of the left texture frame (base view) is performed by using a PMVE, as described in Section 3.3.4.

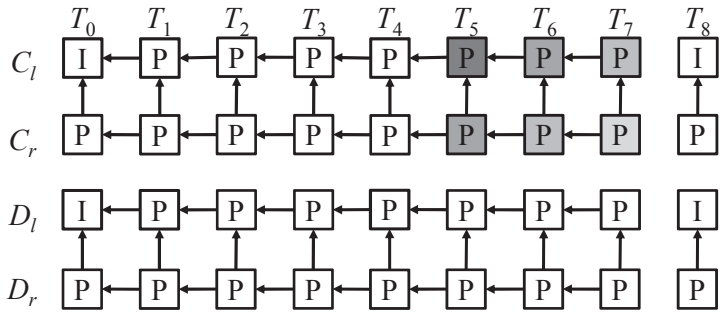



**Figure 3.19** – T. Chang's stereoscopic coding scheme [160].

When a left depth map is lost, it is first verified the availability of the corresponding texture image of the same view. If the texture is available, then MVs are directly copied for the depth map, in order to recover the lost depth frame. If the texture image is also lost, the depth map is recovered by using PMVE [137]. In case the right view is lost (non-base view), then EC is performed by first checking the availability of the corresponding depth map of the same view. If the depth map is available, pixels of the lost right texture frame are synthesised using the depth map associated with the left texture image. If the corresponding depth map is not available, detection of occluded

pixels is performed by using both texture and depth from right view. After detecting the occluded pixels, the non-occluded ones are first concealed by using the inter-view similarities, as described in [146] and in Section 3.4. The remaining occluded pixels are then recovered using PMVE. Finally, the right depth frames errors are concealed directly using MVs from the corresponding right texture frame. For pixels that have been encoded with DCP, the temporal MV is not available and its recovery is done by using the method described in [146]. To evaluate the proposed EC method, T. Chung compared it with four other reference methods described in [142, 158, 159, 161]. Random full frame loss was simulated for both texture and depth maps in both views using only one error percentage (10%). It was considered that the first GOP and the first pictures of all GOPs are always received without errors. Since this EC method is intended to be used on both texture and depth, the author evaluated the PSNR of texture video sequences and also the intermediate synthesised view using the recovered texture and depth maps. When compared with the reference methods, the proposed method is able to obtain consistent PSNR improvements over all tested sequences on both texture images and also over the corresponding synthesised views. Besides the good concealment results it would be enlightening to know what are the negative effects on the synthesis if errors occurred only in depth maps. In a test scenario where texture and depth maps sequences are individually encoded, the accuracy of the EC techniques for each type of data could be more easily assessed.

X. Liu in [162] followed an approach with some points in common with the one described in the previous paragraphs [160]. At the encoder side, the authors start by detecting the occlusions in the texture regions. Then, the texture MVs of such regions are used in the encoding process of the depth. When a frame is lost, at the decoder side the EC is performed in two main steps: first the non-occluded regions of the lost frame are recovered by synthesising those pixels; in the second step, the MVs that were embedded in the coding process are used to reconstruct the lost regions that correspond to the occluded areas. The approach of forcing texture MVs of the occluded regions in the depth maps is an innovative idea, but despite texture MVs being highly correlated with depth maps MVs, as mentioned before, this redundancy tends to decrease in fast moving regions. Since X. Liu uses the texture MVs in the depth maps coding, the author should have measured the cost of using such non-optimal MVs. Since those MVs are not the original ones, the cost is translated by an increased residue due to less accurate motion

information. To evaluate the proposed method, it is computed the PSNR of texture images and depth maps. Regarding the texture images, the author reported significant PSNR advantages over the reference method. In the case of the depth maps, also good PSNR results were reported but, as mentioned before, since depth maps are not used for display its quality should be verified by assessing the quality of the synthesised images using the data resulting from the concealment process.

C. Hewage in [163] proposed an EC for both texture and depth maps for MVD format. Note that this method is based on an SVC encoding scheme, where the texture is encoded as the base layer and the depth maps were encoded as an enhancement layer. Since only one view was used, inter-view correlations are not exploited in this work. The authors rely on the assumption that due to the chosen SVC coding scheme, when a region of the texture is lost, the co-located region of the depth map has a high probability of being also corrupted, due to the inter-layer correlations. The proposed technique starts by recovering errors in the texture images errors. By exploiting the correlations between texture and depth, the depth map is recovered in a second step. This method is based on the assumption that only blocks of frames are randomly lost along the texture image and depth maps. Therefore, besides using temporal information from the temporally neighbouring regions, the spatial neighbouring regions of the lost regions can also be used by a block matching technique, in order to find a more accurate block to reconstruct the lost one. In this technique, proposed by C. Hewage, the lost regions are classified into two separate groups. The first group includes the lost blocks that have at least one surrounding block in the spatially neighbouring regions. The second group includes either blocks with corrupted neighbour blocks or blocks that have already been concealed. In the case of missing regions, classified as belonging to the second group, EC is performed by simply using temporal replacement (TR), using the last decoded frame. In the case of the lost regions belonging to the first group, EC is performed as shown in Figure 3.20. By using the available blocks on top, bottom, left and right of the lost block, a search is performed in order to find four MVs that correspond to these four neighbour blocks. The search is done on the previously decoded frame, which is used as the reference frame with a $32 \times 32$ pixel search window. The best MV is chosen by computing a border continuity metric (BCM), as described in [164]. After choosing the best MV, the lost block is recovered by motion compensation.

Regarding the EC of the depth map, a lost block is also classified into two groups, by

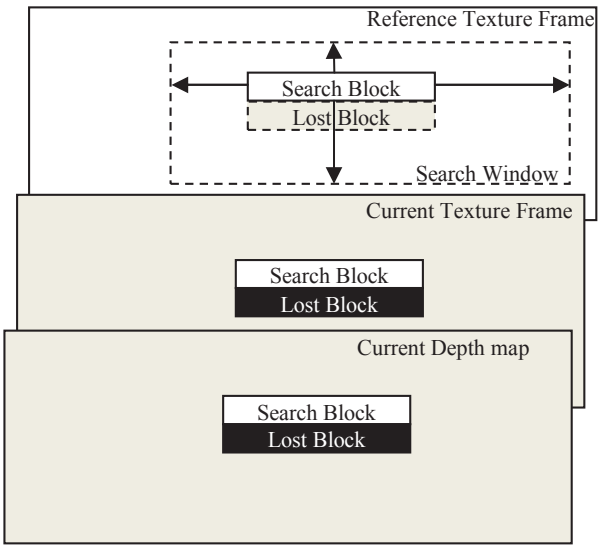



**Figure 3.20** – C. Hewage texture and depth error concealment [163].

determining if the lost region is co-located (first group) or not (second group) with the lost region in the texture image. If the lost depth block belongs to the first group, EC is performed by using the best MV found in texture. The authors rely on the high correlation that exists between texture and the corresponding depth maps. In the case where the lost depth regions belongs to the second group, the EC is also performed by using TR. To evaluate the proposed method, C. Hewage computed directly the PSNR and SSIM of the recovered texture and depth using only one reference method (TR). The authors presented an EC method which shares many points in common with other solutions presented in this chapter, such as performing block matching search in the neighbourhood of the lost regions using the correlations between texture and depth. In this respect, the work is lacking comparison with those methods or, at least, with one temporal EC algorithm more complex than TR, such as MoCp, MVE or PMVE. The authors reported PSNR advantages over TR up to 0.77dB for texture and 0.23dB for depth maps. Regarding the SSIM, the results show that the highest difference between the proposed method and the TR is less than 0.4%, which might reveal that SSIM is not an appropriate metric to evaluate the recovered texture and depth in this scenario, due to the small differences. The authors also did not synthesise any image or video with the recovered MVD sequence, which is relevant to evaluate the accuracy of the depth maps to synthesise virtual views.

W. Lie proposed in [165] an EC method for MVD, which aims to recover entire frame loss

for both texture and depth maps. Regarding the recovery of depth maps, which is the main focus of this section, the authors proposed a technique based on MV sharing [159] and BMA. Typically, BMA is not used in full-frame loss but rather in error concealment of lost regions that have at least some neighbour regions decoded without errors. In this method, W. Lie starts by using MV sharing on the co-located regions of the texture image where there are available MVs. In Intra coded texture MBs, the MB is divided into $4 \times 4$ pixel sub-blocks and, for each of them an MV is computed. These co-located MVs with the texture intra-coded blocks are computed using the texture MVs of the neighbouring region of the intra-coded MB. All the MVs recovered in this step are designated as DEC_BF. The subsequent task, is to refine DEC_BF with BMA using the information from depth map together with the corresponding texture information. The authors assessed the accuracy of the proposed method by separately computing the PSNR of both the recovered texture and depth maps. As mentioned before, using PSNR to measure the quality of depth maps might not be the best performance evaluation. This problem is more evident when the PSNR differences are smaller. In one of the five sequences used by the authors, the depth PSNR advantage over the reference methods is up to 2.42dB for a 15% PLR, but for all the other sequences the depth PSNR gain is lower than 0.5dB. Even for the depth maps with higher PSNR gain, such results may not be valid performance indicators since the corresponding synthesised views were not evaluated.

In [166], X. Zhang proposed an EC technique for depth maps, which is based on the selection of three well known other techniques. The first technique is weighted spatial interpolation of the four uncorrupted neighbours. The second technique is an inter-frame EC method, where a simple TR replacement is used. Finally, the third technique is MV sharing, as described in [159]. The method proposed by X. Zhang starts to recover each missing block using the three methods mentioned above. The winning concealment method is selected by verifying the similarity of the recovered depth block with the corresponding region of the depth maps in the adjacent view of the same temporal instant.

The similarity is based on the computation of the disparity MVs based on the depth maps. Figure 3.21 shows an example of how this task is performed. It assumes that the left view ($I_l$), right view ($I_r$) and the corresponding depth maps ($D_l$ and $D_r$), $k = (x, y)$ define the coordinates of the texture co-located depth block $B$ in the left view, while $k' = (x', y')$ defines the position of the associated block $B$ in the right view. In the non-occluded regions, the original depth maps, the DVs (disparity vectors) $v_l(k)$ and $v_r(k')$ should be

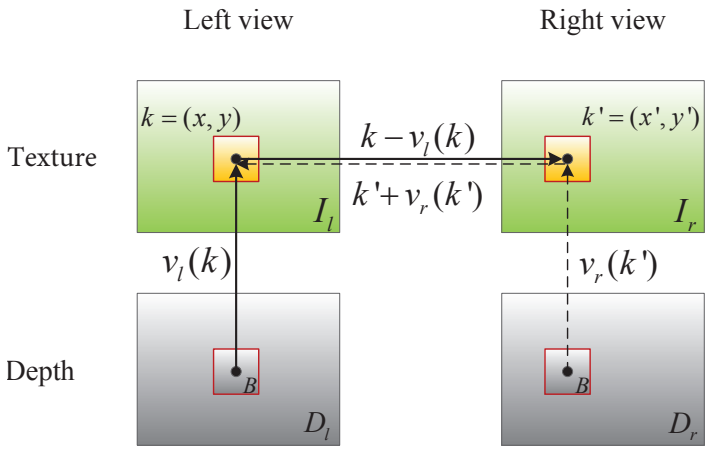



**Figure 3.21** – X. Zhang disparity MVs computation based on depth maps [166].

very similar. These DVs are computed based on the corresponding depth maps, as defined by Equation 3.42. In this equation only the horizontal component is shown ($vx_l$) since it is considered that only horizontal disparity exists due to 1D arrangement of the camera array.

$$vx_l(k) = f.l_t \left[ \frac{D_l(k)}{255} \left( \frac{1}{Z_{min}} - \frac{1}{Z_{max}} \right) + \frac{1}{Z_{max}} \right] \tag{3.42}$$

$f$ defines the focal length, $l_t$ defines the baseline, and 255 is the maximum value for a depth map with 8-bit resolution. $Z_{min}$ and $Z_{max}$ are, respectively, the minimum and maximum depth value of the depth map sequence. The winning concealment technique is the one that minimises the distortion between the blocks in texture images, which are computed based on the disparity vectors $v_l(k)$ and $v_r(k')$.

### Discussion

The authors evaluate the accuracy of the proposed method by computing the PSNR of the synthesised views. This is performed by using the recovered depth maps of three reference EC methods and the proposed method, which is the combination of these three. The method proposed by X. Zhang is quite interesting, since the EC is based on the computation of the disparity vectors using the recovered depth. This type of approach is not common in the literature, but the authors did not use a realistic error pattern scenario for the performance evaluation. The authors encoded the depth maps using H.265/HEVC

and the loss pattern used $8 \times 8$ sized blocks equally dispersed along the depth map. Additionally it is assumed that error-free decoded regions surround every lost block. This type of scenario is very unlikely to occur since the basic coding units of H.265/HEVC are blocks with $64 \times 64$ pixels. Therefore, at least this block size should have been used to define the error patterns. The author considered a loss scenario where very small blocks are lost, where it is much easier to perform block reconstruction. Usually, a simple spatial error concealment using a bi-linear interpolation of the undamaged neighbours (also used as reference method) presents worse performance under high loss rates, when compared to more complex techniques, or even with temporal EC methods. Since the lost blocks considered in this work have a small size, spatial interpolation using bi-linear interpolation has also a very good accuracy and the proposed method is only able to achieve a PSNR gain of 0.22dB over the reference method, even for a high error rate of 20%.

# 4 Spatial and inter-view error concealment for depth maps

In this chapter, new error concealment (EC) techniques specifically tailored for depth maps are proposed. The EC techniques described in the following sections were developed for intra coded or anchor coded depth maps, without using temporal information. To recover lost data, only information from the same temporal instant is used, which means depth values from neighbouring regions correctly decoded in the same frame and from adjacent views. Such depth information may be complemented with texture pixels from the same view and from adjacent views. Five EC techniques are presented in this chapter, each one described in the corresponding sections.

## 4.1 Depth map error concealment using Bézier curve fitting

In this method, depth map contours representing sharp transitions between different depth levels are reconstructed using curve fitting techniques based on Bézier curves [167]. Firstly, all contours representing sharp transitions in depth values are extracted from the received depth map. Secondly, depth lost blocks are classified into two categories: non-edge lost blocks and edge lost blocks. For the non-edge lost blocks, weighted sample interpolation is used to compute values for the missing depth samples, while for edge lost blocks the missing edge/contour is first reconstructed by using Bézier curves. Based on the recovered

depth contours, depth values inside such lost blocks are also computed using weighted interpolation. In this case, contours are used as boundaries to separate regions with different levels of depth. Bézier curves were previously used in image/video EC methods [168–171], but its application in error recovery of depth maps has not been addressed in previous works.

The proposed algorithm comprises four main steps, as shown in Figure 4.1. Initially, all contours are extracted from the depth map. Then, the contour around each lost area is analysed to find matching endpoints that should be connected together. Finally, based on the matching endpoint pairs obtained in the previous step, an additional pair of control-points is computed to reconstruct the contour using a Bézier curve. Finally, all lost blocks are reconstructed using weighted sample interpolation.

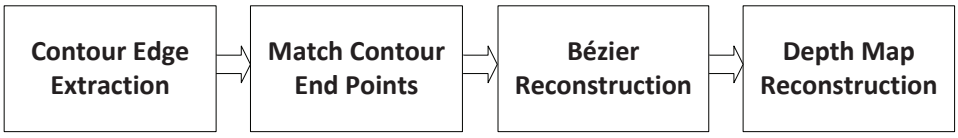



**Figure 4.1** – Depth map error concealment using Bézier curves block diagram.

### 4.1.1 Edge extraction with a variance method

This low complexity contour extraction technique is based on the variance of a sliding window with $3 \times 3$ samples size. Firstly, the variance of such window is computed for the whole image. Then, contours are defined for those depth map samples with a variance above a pre-defined threshold. A value of 5 was empirically obtained, after exhaustive testing, as a good trade-off between the sharpness of depth transitions across edges and the contour relevance in the lost region. Figure 4.2 shows an example of a depth map and its extracted contours.

### 4.1.2 Matching end-points

Since several different contour segments might be located within the lost region, in order to reconstruct them separately, the corresponding endpoints should be obtained beforehand and then matched together in pairs, for reconstructing the lost contour segments. Such a

**Figure 4.2** – Depth map and extracted contour, from first frame of Champagne sequence, 39th camera.

pair of endpoints corresponds to the contour points (i.e. depth values), where the missing region breaks the contour.

Figure 4.3, shows an example of a contour crossing a lost area, where $P1$ and $P4$ are the endpoints of Contour 1 and Contour 2, respectively. $\beta_1$ and $\beta_2$ are the angles between the tangent vectors (at the endpoints) and the straight line $\overline{P1 - P4}$ connecting the two endpoints.

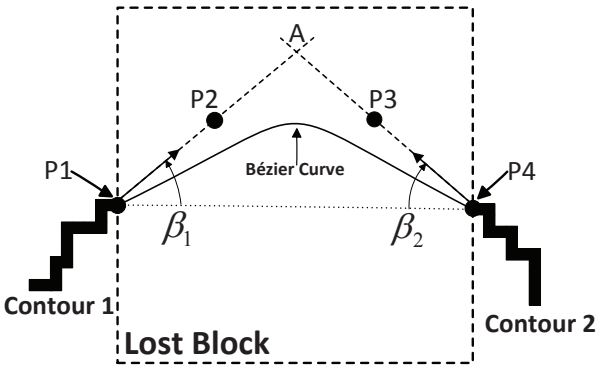



**Figure 4.3** – Match lost contours and Bézier control points.

Reliable and precise computation of $\beta_1$ and $\beta_2$ requires accurate tangent vectors at the endpoints. Equation 4.1 and Figure 4.4 shows an example of how a tangent angle $\beta$ is computed. In this work, up to 5 (N=5) contour depth samples ($P$, $P-1$, ..., $P-N$) and the corresponding $\alpha$ angles ($\alpha_1$, $\alpha_2$, ..., $\alpha_N$), located behind each endpoint, are used to define the tangent angle $\beta$. This was found as an adequate number of points to define

the tangent in the great majority of cases simulated with different depth maps.

$$\beta = \frac{\alpha_1 + \alpha_2 + ... + \alpha_N}{N} \tag{4.1}$$

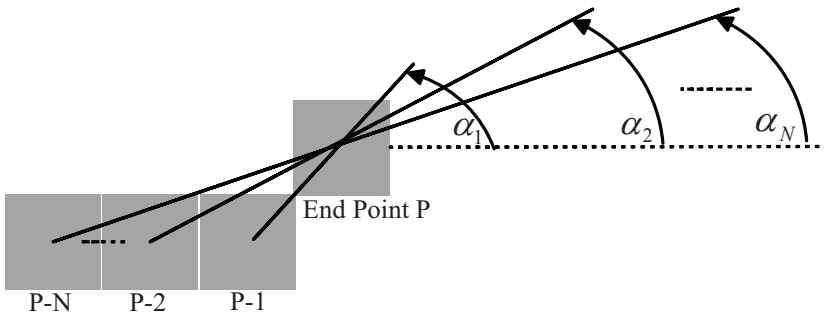



**Figure 4.4** – Tangent angle computation.

Matching the endpoints is accomplished by comparing $\beta_1$ and $\beta_2$ for all possible combinations of two endpoints in each lost region, and then choosing the minimum sum of the corresponding angles $\beta_1$ and $\beta_2$. If some matching pairs give rise to intersecting contours, then such matching pairs will not be considered valid, and another possible matching will be checked. If there is an odd number of end points, the remaining one is not interpolated.

### 4.1.3 Contour reconstruction with Bézier curves

The cubic Bézier curves used to connect the lost contour segments are based on four control-points $P1...P4$, where the first ($P1$) and the last ($P4$) points are the endpoints of the lost contour segment. Control points $P2$ and $P3$ are computed based on the tangent vectors at endpoints $P1$ and $P4$. The first step is to find the control points $P2$ and $P3$ by drawing two virtual lines passing through P1 and P4, with angles $\beta_1$ and $\beta_2$, respectively. Then, A as the intersection point of the two virtual lines (see Figure 4.3), $P2$ and $P3$ are the points located at the middle of segments $\overline{P1-A}$ and $\overline{P4-A}$. Finally, these four control points are used to define the Bézier curve, which in turn is used to reconstruct the lost segment of the broken contour.

A parametric Bézier curve is represented by $Q(t)$, where $t \in [0,1]$ and $P_1...P_4$ are the 4 control-points described above.

$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4 \tag{4.2}$$

The Bézier curves $x(t)$ and $y(t)$ representing $Q(t)$ are:

$$\begin{aligned} x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x && (4.3)\\ y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y && (4.4)\\ & && (4.5) \end{aligned}$$

Defining $C$ as the coefficient matrix of $x(t)$ and $y(t)$, and $F$ as the parameter vector of the parametric curves, i.e.,

$$\begin{aligned} C &= \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \end{bmatrix} && (4.6)\\ F &= \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}^T && (4.7) \end{aligned}$$

$Q(t)$ can also be written in a matrix form as follows:

$$Q(t) = [x(t) \;\; y(t)]^T = C.F \tag{4.8}$$

Since Equation 4.8 does not depend on any control point, in order to find the Bézier curve that represents the lost segment of the depth contour, one has to define an equation with P1...P4. This can be obtained by using the definition of the Bézier curve based on the basis matrix $B$, as defined by Equation 4.9 [172], i.e.,

$$B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.9}$$

$$Q(t) = P_i.B.F \tag{4.10}$$

where the four control-points $P_i$, with $i = 1...4$, are defined by the corresponding coordinates $x_i$ and $y_i$, i.e.,

$$P_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, i = 1, 2, 3, 4 \tag{4.11}$$

After substituting Equation 4.8 in Equation 4.10, the solution of the problem consists on finding the elements of matrix $C$. The result is the following set of equations:

$$\begin{aligned}
a_x &= -P1_x + 3 \times P2_x - 3 \times P3_x + P4_x && (4.12)\\
a_y &= -P1_y + 3 \times P2_y - 3 \times P3_y + P4_y && (4.13)\\
b_x &= 3 \times P1_x - 6 \times P2_x + 3 \times P3_x && (4.14)\\
b_y &= 3 \times P1_y - 6 \times P2_y + 3 \times P3_y && (4.15)\\
c_x &= -3 \times P1_x + 3 \times P2_x && (4.16)\\
c_y &= -3 \times P1_y + 3 \times P2_y && (4.17)\\
d_x &= P1_x && (4.18)\\
d_y &= P1_y && (4.19)
\end{aligned}$$

After computing matrix C, the parametric Bézier curves $x(t)$ and $y(t)$ are completely defined. Then, the interval of t ([0,1]) is divided by twice the length of segment $\overline{P1 - P4}$ (measured by the number of depth samples), in order to obtain the number of points of the Bézier curve. Finally, based on the computed curve, the closest depth pixels inside the lost region are chosen to build the reconstructed contour.

### 4.1.4 Depth map reconstruction

After recovering all lost contour segments, the missing depth values within a lost region must be recostructed. As mentioned before, every lost block is classified based on the existence of a contour segment. The first category includes those blocks that have contours crossing them (i.e., broken contours), while the other category includes the remaining blocks.

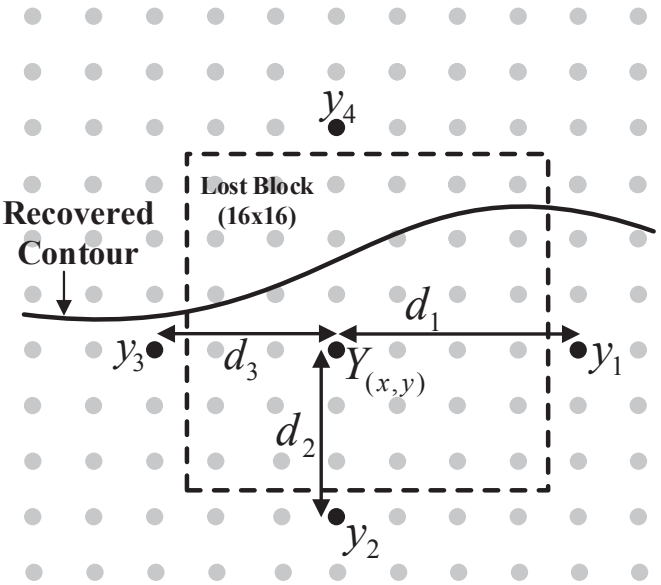



**Figure 4.5** – Weighted interpolation of the depth values of the lost block.

Figure 4.5 shows the reconstruction process of a missing depth value $Y_{(x,y)}$ inside a lost block, by using weighted interpolation based on four depth values ($Y_i, i = 1...4$) located in the boundaries of the lost block. Depending on the category of the lost block, not all adjacent depth values can be used. If a lost contour segment is reconstructed across a block, then there are adjacent depth values located on opposite sides of the contour that cannot be used, because they belong to distinct depth levels. For instance, in Figure 4.5 pixel $Y_4$ is not used for interpolating $Y_{(x,y)}$ because it is located on the opposite side of the contour, thus most likely belongs to a very different depth plane. The weighted interpolation is implemented as defined by Equation 4.20:

$$Y(x,y) = \frac{\sum_{i=1}^{N} Y_i \times [15 - d_i]}{\sum_{i=1}^{N} d_i}, \tag{4.20}$$

where $d_i$ is the depth value distance between $Y_{(x,y)}$ and $Y_i$, which can have a maximum value of 15 for a 16x16 block size.

### 4.1.5 Experimental results

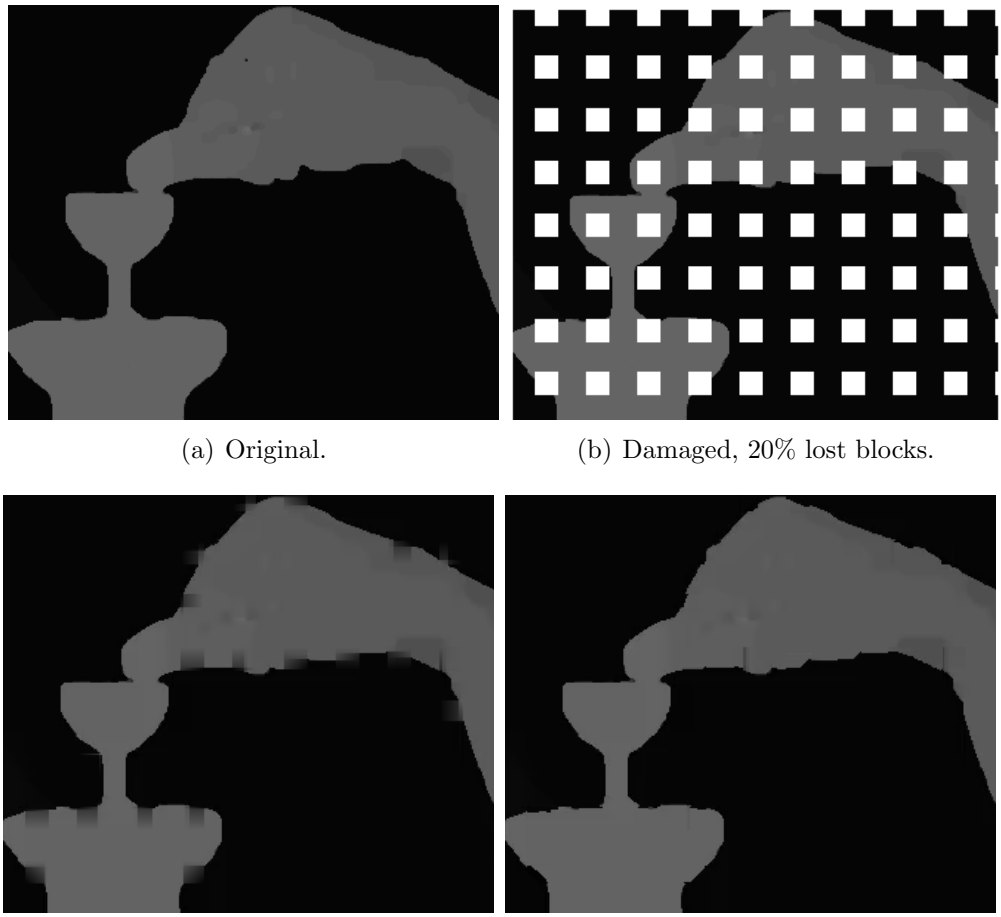
The performance of the EC method described in the previous section was evaluated by using the first frame of the five sequences presented in Table 4.2. Since the proposed method is a spatial EC algorithm, any single frame could be used in the experiments. The texture images and their corresponding depth maps were encoded using H.264/AVC intra mode and fixed QP=28. The reference H.264/AVC software JM17 was used [173]. The PSNR obtained for texture and depth is shown in Table 4.2. These texture images and depth maps were used to evaluate the quality of the synthesised view obtained by using the recovered depth map. The VSRS (View Synthesis Reference Software) of the MPEG group was used to synthesize the virtual view [29].

**Table 4.1** – Tested sequences (dB).

| Sequences | Resolution | Texture (PSNR) | Depth (PSNR) |
|---|---|---|---|
| Ballet | 1024×768 | 41.04 | 46.25 |
| Breakdancers | 1024×768 | 40.05 | 48.08 |
| Book Arrival | 1024×768 | 40.50 | 45.89 |
| Champagne | 1280×960 | 43.28 | 49.60 |
| Beergarden | 1920×1080 | 38.76 | 48.22 |

A regular error pattern was used in the entire image, in order to obtain a large number of different types of lost contour segments. Thus, depth map losses are not constrained within any particular type of image region. The error pattern was defined as 16×16 lost blocks equally spaced, as shown in Figure 4.6(b)). This type of error may result from transmission errors (namely burst errors) in video streams coded with Flexible Macroblock Ordering (FMO) [115].

The simulation tests were done for three different block loss rates: 5%, 10% and 20%. Performance evaluation was carried out by measuring the PSNR of the synthesised views using the recovered depth maps. A spatial EC method based on weighted interpolation only, without taking into account the depth map contours, is used as reference for comparison (Ref.) with the proposed one (Prop.). Since the inpainting process for occlusions of VSRS shows a major impact on the PSNR [174], the occluded regions were not included in the PSNR computation of the synthesised view. Therefore, the PSNR obtained for these views is exclusively due to the performance of the depth error recovery method, and is not influenced by the inpainting process.



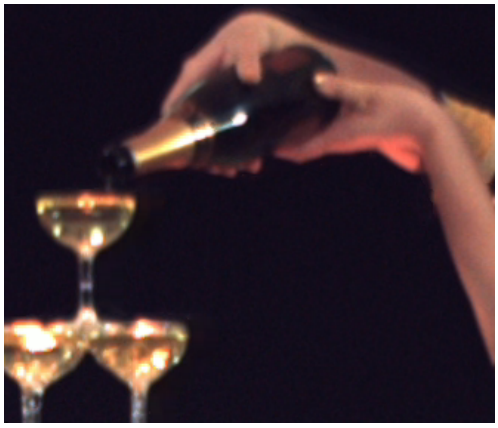
(a) Original. (b) Damaged, 20% lost blocks.

(c) Recovered with ref. method. (d) Recovered with prop. method.

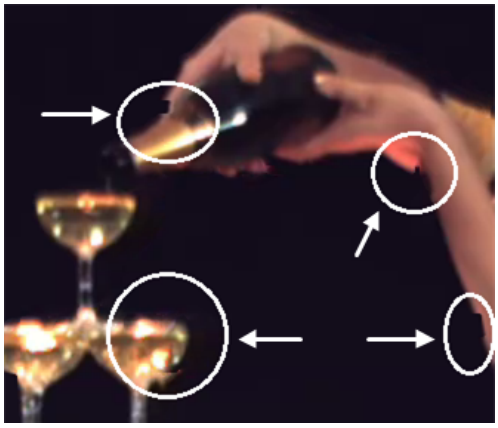
**Figure 4.6** – Recovered Depth Maps (20% lost blocks).

Table 4.2 shows the PSNR of the synthesised views for the test sequences, at 5%, 10% and 20% block loss rates. As shown in this table, the proposed method outperforms the reference method for all sequence. These results highlight that PSNR gains of the proposed method are generally higher for larger loss rates, proving its effectiveness.

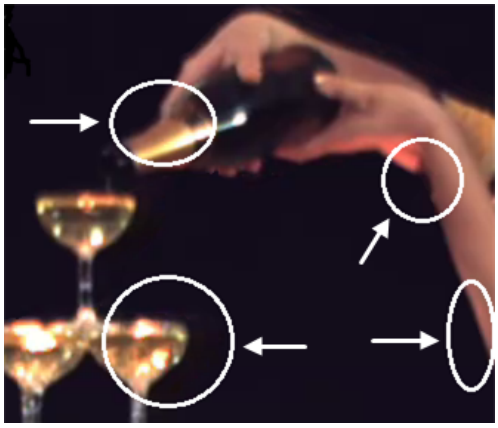
These results can be subjectively confirmed by observing some image details shown in Figures 4.7(a) to 4.7(c). A detailed observation of the synthesized view indicates that regions reconstructed with the proposed method (Figure 4.7(c)) are much more accurate than in Figure 4.7(b). This is because the proposed method first reconstructs the lost edges of the depth map, which are taken into account for the sample interpolation. Note that, these regions correspond to sharp depth transitions with significant impact on the perceived 3D quality. Moreover, the block effect that is visible in reference method is almost vanished, when the proposed method is used.



(a) Original.



(b) Reference method.



(c) Proposed method.

**Figure 4.7** – Synthesized Views (20% lost blocks).

**Table 4.2** – Results for the synthesised images (dB).

| Seq. | No errors | 5% Loss | | 10% Loss | | 20% Loss | |
|---|---|---|---|---|---|---|---|
| | | Ref. | Prop. | Ref. | Prop. | Ref. | Prop. |
| Ballet | 36.07 | 34.85 | 35.55 | 34.77 | 35.89 | 33.93 | 35.64 |
| Breakdancers | 38.93 | 38.86 | 38.87 | 38.81 | 38.87 | 38.28 | 38.69 |
| Book Arrival | 40.07 | 39.73 | 39.88 | 39.51 | 39.74 | 38.11 | 38.64 |
| Champagne | 39.58 | 37.97 | 39.29 | 36.67 | 39.17 | 36.11 | 38.02 |
| Beergarden | 34.60 | 34.15 | 34.39 | 33.83 | 34.22 | 33.58 | 34.03 |

## 4.2 Depth map error concealment using texture image contours

The method described in this section is based on a different approach from the one described in Section 4.1. In this approach, each corrupted depth map is firstly processed to obtain differences between depth planes, thus defining relevant edges (or contours). Afterwards, object boundaries of the texture image are extracted and represented as contours in a binary map. The similarity between the texture contours and those of the depth map is exploited, and the corrupted depth contours are restored using information from the texture image. Finally, the corrupted blocks of the depth map are interpolated using the geometric information inferred from the restored contours. This method is also used to accurately recover most shapes present in the depth map, achieving high quality synthesized views.

### 4.2.1 Exploiting similarities between depth maps and texture

In this type of techniques, recovering the missing blocks is based on the assumption that the corresponding texture image is decoded without errors. Additionally, the performance of this method is evaluated based on the quality obtained in the virtual view synthesised from the recovered depth map and the corresponding texture image.

This EC method comprises three main stages, as represented in the block diagram of Figure 4.8: (1) contour extraction of both texture image and depth map; (2) depth contour reconstruction in missing blocks, based on the texture image, and (3) weighted interpolation of the missing depth values, using the reconstructed contours as arbitrary region boundaries in the interpolation process. In the second stage contour reconstruction of the depth map comprises two steps: firstly, the correlation between contours located in the surrounding area of the lost region in the depth map and the corresponding area in the texture image is exploited; secondly, the most correlated texture image contours are used to recover the missing contours in the depth map.

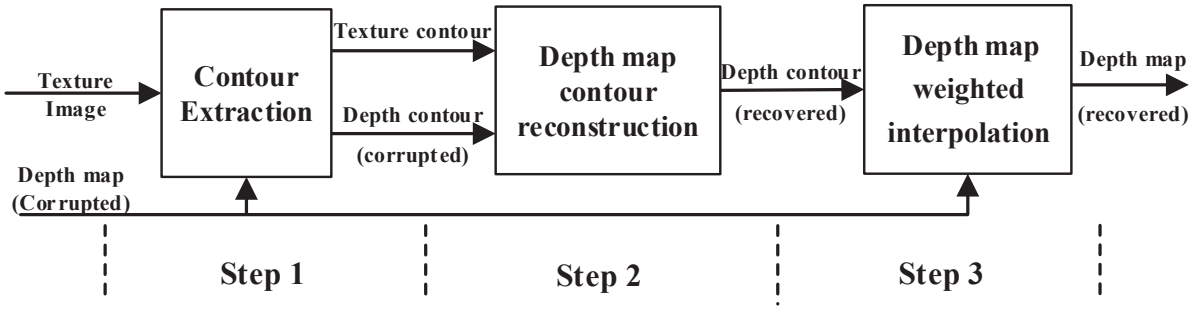



**Figure 4.8** – Proposed method: processing stages.

### 4.2.2 Contour extraction - Canny edge detection

In this EC method, contour extraction is obtained throughout an algorithm that is different from the one described in Section 4.1.1. The variance presents a good performance with depth maps because these are essentially composed of smooth regions with sharp and well-defined edges. When extracting contours from a texture image, the performance of such edge extracting technique is not satisfactory because texture images have significantly more complex content, and the appearance of false edges may occur. Therefore, to minimize this problem, an edge extraction method based on the Canny edge extraction technique is used [112].

The *Canny edge detector* algorithm is used to extract the contours of both texture image and depth map based on the implementation made available by the authors in [175]. This method comprises five steps: The first step consists in filtering the input image $I(x, y)$ (texture and depth map), using a Gaussian filter with $\sigma = 1$, window size of $9 \times 9$. Such filtering, described by Equation 4.21, aims to reduce the noise by smoothing the image, in order to minimise false edge detection.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} \tag{4.21}$$

The second step computes the gradient of the smoothed image along the $x$ and $y$ axes, i.e., $dx(x, y)$ and $dy(x, y)$. This is done by convolving the image with the first partial derivative of the Gaussian function (Equation 4.21) with respect to $x$ and $y$, in order to obtain $dx(x, y)$ and $dy(x, y)$, respectively.

In the third step, the gradient magnitude $mag(x, y)$ is computed using the Equation 4.22

and its direction $\phi_{x,y}$ at every pixel, where $\phi$ is defined by Equation 4.23.

$$mag(x,y) = \sqrt{(dx(x,y))^2 + (dy(x,y))^2} \tag{4.22}$$

$$\phi_{x,y} = tan^{-1}(dy(x,y)/dx(x,y)) \tag{4.23}$$

In the fourth step, given the computed gradients and the respective angles, the non-maximum suppression is executed to obtain the first edge map, usually called the thin edge map, which is already an edge map, though rougher than the final one.

Finally, in order to obtain the final edge map, a hysteresis threshold is used for the thin edge map, where the high and low threshold levels of the hysteresis range are computed based on a 64 level histogram of the previously computed magnitude $G(x,y)$ (the magnitude is normalized to the interval [0;1]). In order to obtain the threshold, the following conditions are assumed: (a) only a maximum of 30% of the edge map values are considered as valid edge samples; (b) after computing the highest threshold, it is assumed that the low threshold is half of the higher one. Based on these conditions, the histogram is read by scanning the number of samples of each sub-interval, starting from the value 0 to 1. After each scan, the number of hits is accumulated into the variable $Sum_{hits}$. When $Sum_{hits}$ overpasses 70% of the total number of pixels of the edge map, the histogram scanning is finished. The value of the sub-interval that was last checked is chosen to be the highest threshold. Figure 4.9 illustrates the result of edge extraction using this method.

### 4.2.3 Contour reconstruction

In the process of contour reconstruction, small depth contour segments around the missing area are not considered, as they usually do not represent significant changes in depth, when compared to the size of the missing area. Then, all the contour endpoints around the border of the lost area are scanned, and every endpoint that belongs to a contour segment with less than three pixels is not considered for reconstruction.

As mentioned before, after determining the contour endpoints of the lost blocks, the next step is to reconstruct the lost contour segments, by exploiting the correlation between the texture image and depth map contours in the corresponding image areas. Figure

**Figure 4.9** – Example of extracted contours: Texture image (light gray), depth map (dark gray), overlapped (black).

4.9 shows an image area where the contours of the texture image overlap with those on the depth map. As can be seen in Figure 4.9, there are clear similarities between both contours, that can be efficiently exploited.

The use of the texture image contours to recover the corresponding lost segments of the depth map is exemplified in Figure 4.10. The first step is to search for a possible contour match, in the corresponding surrounding region of the texture image, for each endpoint of the depth map contour. In Figure 4.10, C1 and C2 are possible matches for depth map contour endpoints D1 and D2, respectively. Secondly, for each possible match, the continuity of the texture image contour inside the lost area is checked; if it is possible to connect two endpoints (in this case C1 and C2), then the texture image contour will be used to reconstruct the lost segment of the depth map. If there are multiple possible matches to reconstruct the lost contour, then the best candidate is found through the following steps: (1) all angles of the tangent vectors pointing to the lost block are calculated ($\alpha 1$, $\alpha 2$, $\beta 1$ and $\beta 2$), and, (2) the best match will be selected as the contour of the texture image, whose angle is the closest to that of the lost depth map contour ($\alpha 1$ and $\alpha 2$).

In the example of Figure 4.10, texture contour C1-C2 is copied into the damaged depth

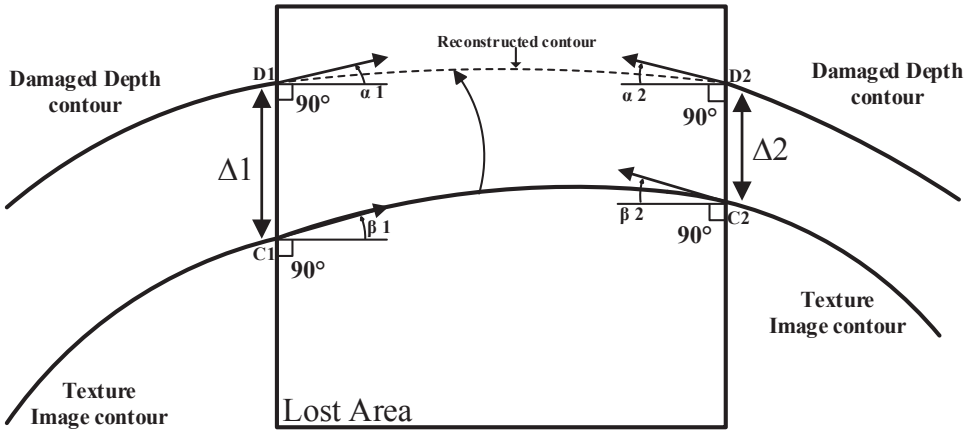



**Figure 4.10** – Matching texture image contours to depth map.

contour D1-D2, since they are not exactly coincident. This process involves calculating the distances between the depth map contour endpoint and the corresponding points from the texture contour ($\Delta 1$ and $\Delta 2$). Then the texture image contour will be shifted by a distance that is the average of $\Delta 1$ and $\Delta 2$. After this, if the reconstructed contour does not match exactly the depth map endpoints, then it is directly connected to the closest endpoint (usually this is not more than 2 pixels).

### 4.2.4 Depth map reconstruction

After recovering all missing contour segments, the missing depth values of the lost region in the depth map must be reconstructed. Every lost block is classified according to the existence of contour segments inside a lost area. The first category includes blocks with crossing contours, while the other category includes the remaining ones. Interpolation of the lost values is performed using the method described in Section 4.1.4.

### 4.2.5 Experimental results

The performance of the proposed method was evaluated by using the first frame of five sequences, presented in Table 4.3. The sequences and test scenario is the same as that used in Section 4.1, where the same encoding and loss environment is used.

The performance is evaluated by comparing the quality of the synthesized views obtained

after decoding the depth maps and texture images. The depth map without data blocks establishes the error free quality, while recovered depth maps using the proposed and reference methods correspond to the case where corrupted depth maps are received. The performance of the different methods used to recover the missing blocks in depth maps is then evaluated by computing objective quality metrics for the corresponding synthesized views. The results are shown in Table 4.3, using PSNR and SSIM (structural similarity) [176], for the block error rates previously mentioned. The differences (between brackets) result from the comparison with the error free case, i.e., synthesised view using a depth map decoded without errors.

These results show that the proposed method is able to outperform the reference one, especially for images with well-defined objects in the texture image, where the extracted contours are very useful. This is the case of image *Ballet*, in the worst case scenario, where for losses of 20% in the depth map the proposed method only achieves 0.37dB less than *No Errors* case, and the reference method has a loss of 1.86dB relative to the reference method. The lowest gain occurs for *Breakdancers* at lower error rates (5% and 10%). This is because in very homogeneous depth map regions, simple weighted average interpolation is almost as good as the proposed method. However, when the error rate increases and the lost blocks affect a higher number of regions with different levels of depth, i.e., with crossing edges, the proposed method provides better quality due to the advantage of reconstructing the depth map contours, thus improving the weighted interpolation of the missing values.

These objective results can be subjectively assessed by observing Figure 4.11, where the synthesized images and the respective depth maps are shown. It is possible to observe that in regions where depth maps are reconstructed with higher quality, the synthesized image also presents a higher subjective quality. This can be noticed, for instance, in object boundaries with large variation in the depth values and well-defined objects in the texture image (see the highlighted regions in the Figure 4.11). In these cases, the proposed method is able to efficiently reconstruct the lost areas.

This novel method reconstructs missing blocks of depth maps, by combining information from both adjacent regions of the corrupted depth map itself and co-located areas of the respective texture image. Experimental results show that the proposed method is able to efficiently reconstruct corrupted depth maps by using the recovered contours as region

**Table 4.3** – Experimental image synthesis results (PSNR(dB) and SSIM[0-100]).

| | | | Reference Method | | | Proposed Method | | |
|---|---|---|---|---|---|---|---|---|
| | Sequence | No Errors | 5% Err. | 10% Err. | 20% Err. | 5% Err. | 10% Err. | 20% Err. |
| PSNR | Ballet | 35.55 | 34.47 (-1.08) | 34.38 (-1.17) | 33.69 (-1.86) | 35.49 (-0.06) | 35.36 (-0.19) | 35.18 (-0.37) |
| | Book Arrival | 40.07 | 39.73 (-0.34) | 39.51 (-0.56) | 38.12 (-1.95) | 39.75 (-0.32) | 39.70 (-0.37) | 38.68 (-1.39) |
| | Breakdancers | 38.93 | 38.86 (-0.07) | 38.82 (-0.11) | 38.3 (-0.63) | 38.86 (-0.07) | 38.81 (-0.12) | 38.72 (-0.21) |
| | Champagne | 39.58 | 37.98 (-1.60) | 36.74 (-2.84) | 36.23 (-3.35) | 38.02 (-1.56) | 37.37 (-2.21) | 36.61 (-2.97) |
| | Beergarden | 34.60 | 34.14 (-0.46) | 33.83 (-0.77) | 33.59 (-1.01) | 34.23 (-0.37) | 33.92 (-0.68) | 33.72 (-0.88) |
| SSIM | Ballet | 91.96 | 91.64 (-0.32) | 91.31 (-0.65) | 90.95 (-1.01) | 91.81 (-0.14) | 91.59 (-0.37) | 91.36 (-0.60) |
| | Book Arrival | 94.53 | 94.48 (-0.05) | 94.40 (-0.13) | 94.16 (-0.37) | 94.49 (-0.04) | 94.43 (-0.10) | 94.26 (-0.27) |
| | Breakdancers | 92.32 | 92.29 (-0.03) | 92.22 (-0.10) | 92.08 (-0.24) | 92.28 (-0.04) | 92.21 (-0.11) | 92.16 (-0.16) |
| | Champagne | 97.10 | 97.59 (-0.12) | 97.44 (-0.27) | 97.34 (-0.37) | 97.60 (-0.11) | 97.46 (-0.25) | 97.36 (-0.35) |
| | Beergarden | 94.84 | 94.70 (-0.14) | 94.60 (-0.24) | 94.37 (-0.47) | 94.70 (-0.14) | 94.61 (-0.23) | 94.42 (-0.42) |

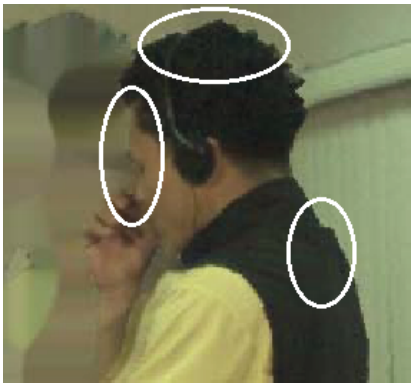
boundaries in the weighted interpolation of the missing values. The synthesized images, using such reconstructed depth maps, achieve higher quality than those using simple weighted interpolation to recover missing depth values up to 1.49 dB in the case of *Ballet* for losses of 20%. Overall, the superior performance of the proposed method demonstrates that depth map contours can be efficiently recovered using information from uncorrupted texture images to increase the quality of the depth maps, and thus the synthesized views.

(a) Original.

(b) Damaged.

(c) Reference method.

(d) Proposed method.



(e) Original.

(f) Reference method.

(g) Proposed method.

**Figure 4.11** – Recovered depth maps with the respective synthesised views (20% of block loss).

## 4.3 Depth map error concealment using combined contour reconstruction with Bézier curves and texture information

In this section, a different method for spatial error concealment of depth maps using associated information from texture images is presented. The novel aspects of this method extend the work previously described, providing a hierarchical model comprising two processing stages. The processing burden in each stage depends on the concealment success of the previous one in the recovery of depth map lost contours, which are used to interpolate lost depth map values. These two stages are based on the error concealment techniques described in Sections 4.1 and 4.2.

The flow diagram of this method is depicted in Figure 4.12. Firstly, the edges of both texture image and corrupted depth map are extracted using the *Canny* edge detection algorithm, as described in Section 4.2.2. Secondly, *Stage* 1 is responsible for recovering the contours of the missing area in the depth map by using the corresponding texture image contours; Thirdly, if depth contour segments were not completely recovered in *Stage* 1, *Stage* 2 is responsible for interpolating the broken depth contours using geometric fitting based on Bézier curves; Finally the missing values in the depth map are recovered based on a weighted interpolation using the error-free neighbour values, and the recovered depth contours.

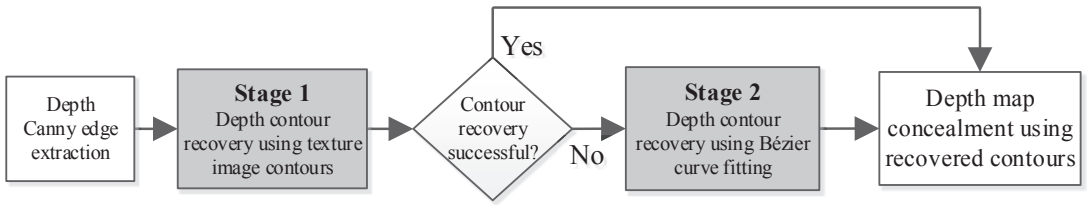



**Figure 4.12** – Proposed algorithm diagram.

### 4.3.1 Experimental results

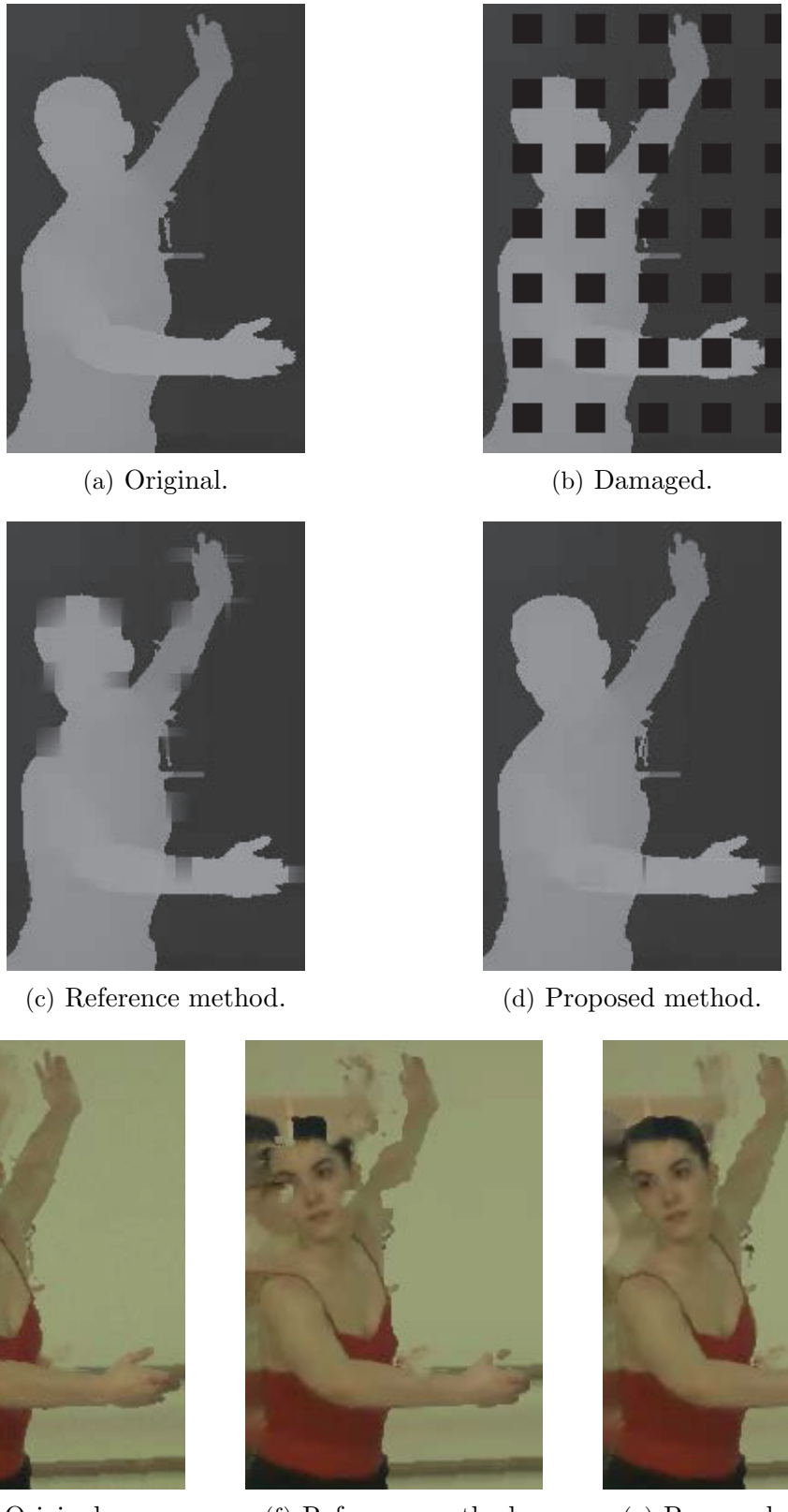
The performance of this EC method was evaluated by using the first frame of the five sequences presented in Table 4.4, using the same test scenario of the methods described in Sections 4.1 and 4.2. The performance of the proposed method was evaluated using

the same data set and also the same test conditions as the ones described in Section 4.2.5 and 4.3.1. For practical reasons, in this experiment the Canny edge extraction algorithm was used as described in Section 4.2.2, but with the implementation available in *OpenCV 2.1* [177], obtaining the necessary contours from texture and depth maps. The low($th_{low}$) and high($th_{high}$) thresholds used by the *Canny* edge detection algorithm are computed based on the *mean* of the image that is being analysed. First the *mean* of the image is computed and then the low and high thresholds are defined as $th_{low} = 0.5 * mean$ and $th_{high} = 1.5 * mean$, respectively. These thresholds were found to be suitable for all texture and depth maps used in the experiments, providing most of the significant contours.

Simulation tests were performed for three different block loss rates: 5%, 10% and 20%. Performance evaluation was carried out by measuring the PSNR of the synthesised views using the recovered depth maps. A spatial concealment method only based on weighted sample interpolation, without taking into account the depth map contours, is used as reference for comparison (Ref.) against the proposed one (Prop.).

Table 4.4 shows the PSNR obtained from synthesised views for the test sequences, at 5%, 10% and 20% block loss rates. As shown in this table, the proposed method outperforms the reference method for all image sequences. Also in this method, the PSNR gains in general increase for higher loss rates, which proves its effectiveness.

These results can be subjectively confirmed by observing the image details shown in Figure 4.13 for the synthesised view, obtained with the proposed method, Figure 4.13(g)); one can clearly perceive that indicated regions are much better reconstructed than in Figure 4.13(f)). This is because the proposed method first reconstructs the lost edges of the depth map and then the sample interpolation, which takes into account the boundaries defined by these well reconstructed edges. Note that these regions correspond to sharp depth transitions with significant impact on the perceived 3D quality. Moreover, by using the proposed method, the block effect that is visible in the reference method is almost eliminated. These results also outperform those obtained in our previous work [10, 11]; depending on the visual content, an improvement of up to 1.01 dB was achieved.



(a) Original. (b) Damaged.

(c) Reference method. (d) Proposed method.

(e) Original. (f) Reference method. (g) Proposed method.

**Figure 4.13** – Depth maps and the respective synthesised views (20% of block loss).

**Table 4.4** – Experimental image synthesis results (PSNR(dB)).

| | | | Reference Method | | | Proposed Method | | | Δ(Proposed-Reference) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sequence — Err. | 0% | 5% | 10% | 20% | 5% | 10% | 20% | 5% | 10% | 20% |
| PSNR | Ballet | 35.55 | 34.47 | 34.38 | 33.69 | 35.47 | 35.32 | 35.35 | 1.00 | 0.94 | 1.66 |
| | Book Arrival | 40.07 | 39.73 | 39.51 | 38.12 | 40.00 | 39.91 | 39.11 | 0.27 | 0.40 | 0.99 |
| | Breakdancers | 38.93 | 38.86 | 38.82 | 38.30 | 39.89 | 38.84 | 38.82 | 0.03 | 0.02 | 0.52 |
| | Champagne | 39.58 | 37.98 | 36.74 | 36.23 | 38.02 | 38.68 | 37.62 | 0.04 | 1.94 | 1.39 |
| | Beergarden | 34.60 | 34.14 | 33.83 | 33.59 | 34.41 | 34.16 | 34.14 | 0.27 | 0.33 | 0.55 |

## 4.4 Depth map error concealment using disparity information

In this section a method for spatial error concealment of depth maps associated with multiview images is presented. The novel aspects of the proposed method include a hierarchical algorithmic structure with three spatial processing stages, each one corresponding to the use of different type of data available at the MVD decoder and associated with the corrupted depth map, i.e., adjacent texture views and depth maps, associated texture view and adjacent regions of the corrupted depth map. In general, the proposed method starts by recovering depth contours in lost regions and then uses them as boundaries to define different weights in bidirectional interpolation. The results show that synthesising views using depth maps recovered by the proposed method, consistently leads to better quality than using weighted interpolation without considering contour boundaries.

The algorithm is structured as a hierarchical three-stage processing method, where each stage is defined by the type of data used in the spatial EC of missing regions in depth maps. Figure 4.14 shows three cases that may occur in MVD, when a corrupted depth map is received and no temporal information is used: (i) reliable data from both adjacent texture images is available; (ii) the only available reliable data comes from the texture image associated with the corrupted depth map; (iii) the texture images do not provide any useful information for concealment of the lost region in the corrupted depth map. Whenever the texture data from texture images is located in image regions where the disparity cannot be computed with good accuracy, this is classified as a non-reliable data region to be used by the concealment algorithm. This is possible to occur in three cases: 1) missing texture data due to losses; 2) occluded regions in the texture view; 3) due to algorithmic inefficiency, the texture disparity tested in the border regions of the lost area in the depth map does not produce a good enough match between both depth maps.

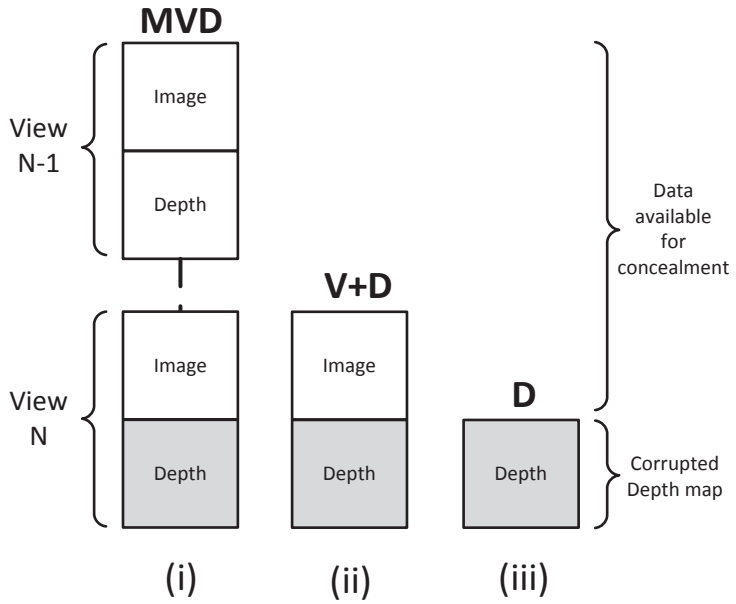



**Figure 4.14** – Depth loss cases in spatial MVD, characterised by the data available for error concealment.

Taking into account these three cases, an algorithmic structure is proposed for spatial error concealment of depth maps, comprising the 3-stages shown in Figure 4.15. *Stage 1* corresponds to case (i), where reliable data from the two views is used for concealment of lost areas in the corrupted depth map. At this stage, the disparity between available views is used to retrieve missing depth values from the depth map of the adjacent view, i.e., disparity compensation between depth maps.

If concealment of all lost areas of the depth map is not fully accomplished in *Stage 1*, because the auxiliary view/depth cannot provide useful disparity information, it means that only one view plus depth is available. Then, this is processed in *Stage 2*, corresponding to the case (ii) in Figure 4.14. In *Stage 2* the contours of the lost regions on the corrupted depth map are first reconstructed by using information from its associated texture image. If all depth map contours of interest are not fully recovered after *Stage 2*, because they were not found in the texture view, for instance, then the algorithm proceeds to *Stage 3*, where only depth data of the corrupted depth map itself is used for concealment, i.e., case (iii) in Figure 4.14. At this Stage, the remaining lost contour segments are reconstructed using Bézier curves. A similar method for contour reconstruction was used in [169], though in a different context. Finally, the lost regions of the depth map are recovered using selective weighted interpolation, based on the reconstructed contours. In the next subsections each stage of the proposed algorithm is explained in detail.

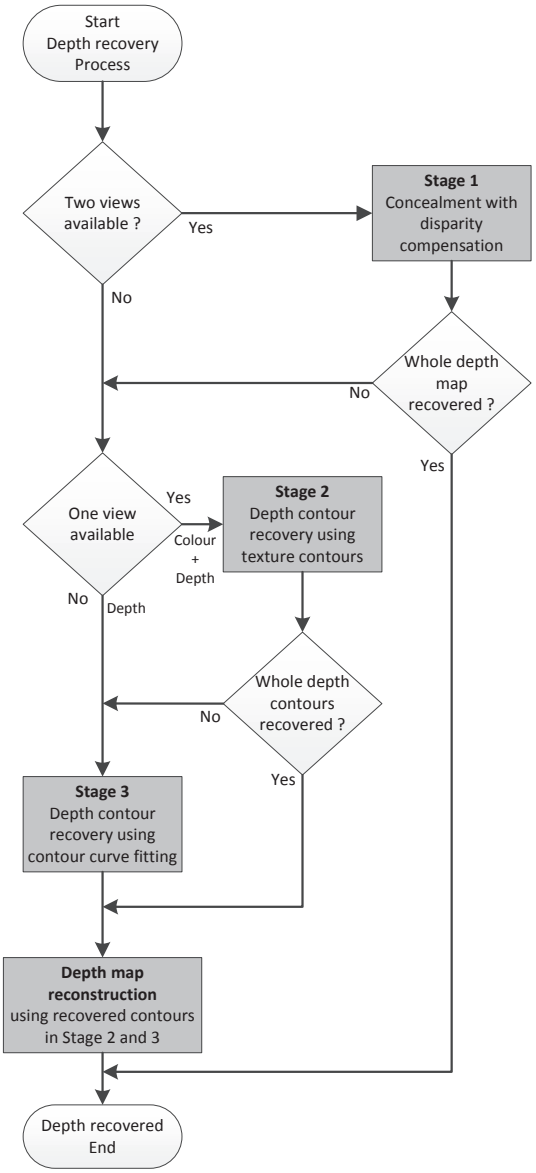



**Figure 4.15** – Algorithmic structure of the error concealment method.

### 4.4.1 Stage 1 - Depth recovery using disparity compensation

Two views (*N-1* and *N*) are used in this processing stage, each one comprising a texture image and the corresponding depth map (see Figure 4.14). It is assumed that *Depth N* is corrupted, i.e., with missing regions. The disparity $D_N$, computed from texture images $N$-1 and $N$, is used to find the best matching depth values in *Depth N-1* to recover the lost ones in *Depth N*. When *Depth N* is fully lost, *Stage 1* is the only processing stage of the algorithm that is used to recover the whole depth map. In this case, *Depth N-1* is

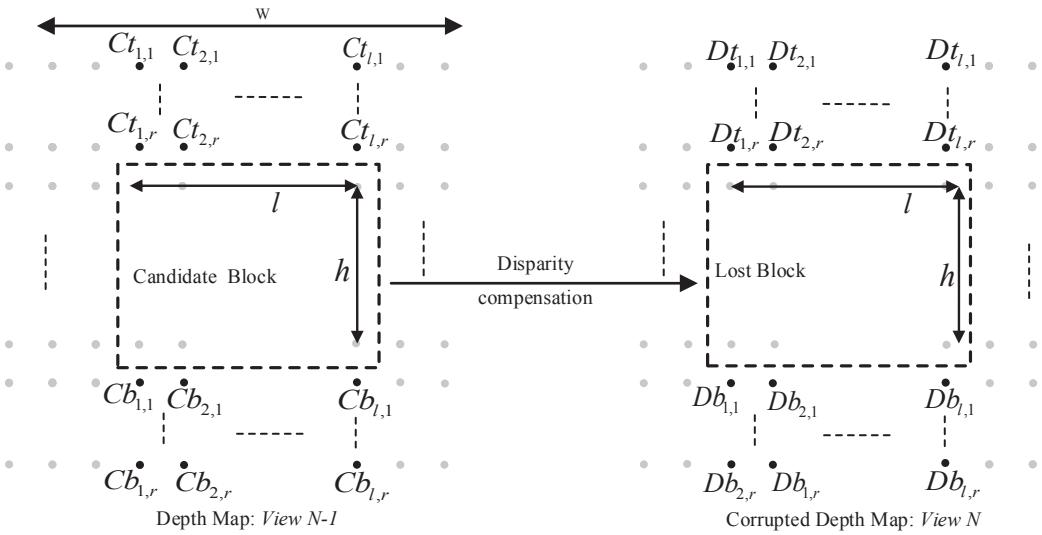



**Figure 4.16** – Depth map block matching using disparity.

used as a substitute of *Depth N* after disparity compensation. Occlusions are filled with gradient-based linear interpolation. The following steps are carried out at *Stage 1*.

### Disparity computation

The first step is to compute a disparity map $D_N$ between *Image N-1* and *Image N*. The disparity computation method used in this work is an algorithm based on a multi-level adaptive method combined with a multigrid technique [178]. It presents low implementation complexity and good efficiency in comparison with others, as demonstrated in the study carried out with the *Middlebury Stereo Evaluation* dataset [179].

### Disparity compensation

The second step in *Stage 1* is to recover lost depth values using disparity compensation. Assuming the standard block-based coding, each lost region corresponds to either a single block or a group of blocks. Each missing region in the depth map is reconstructed by using the disparity compensated values obtained from the corresponding region in the error-free map of the adjacent view.

Since disparity is computed from the texture images, *view N-1* and *view N*, the resulting disparity map is not ground truth data and does not ensure the most accurate values for filling the corresponding lost regions in the corrupted depth map. Therefore, to ensure

the best match between depth maps, the disparity values are further refined before being used to retrieve the depth values from view *N-1* to *N*. The refinement process devised to compute suitable disparity values to recover the missing block is based on the following method. The disparity values corresponding to the lost region are quantised using a fixed number of intervals, ranked in increasing order. In this work 15 quantisation intervals were used; this number of intervals were found through several experiments to be enough to define the initial search points that are used to find the best match. Then, each quantised value is used to define an initial search point for finding the best matching block in *Depth N-1* that should be used to recover the corresponding corrupted one in *Depth N*. The refinement is applied over a range of 4 samples, using the minimum Sum of Absolute Differences (SAD), as the best matching criterion. In this process, the searching area used for finding the best match only includes the depth values belonging to the boundaries of the lost region, because the region itself is missing.

Figure 4.16 illustrates this process using a single lost block in a corrupted depth map. The neighbour depth values of the Candidate Block from a depth map *N-1* are used to find the best match for the Lost Block in *Depth N*, i.e., the top and bottom neighbour values of the lost block, $Dt_{i,j}$ and $Db_{i,j}$, respectively, are used for searching the minimum SAD with $Ct_{i,j}$ and $Cb_{i,j}$. In Figure 4.16, $h$ and $l$ are the height and width of both the lost and candidate blocks, and W is the width of the search window. The best match is found by Equation 4.24, using three rows from the top and another three from the bottom (i.e., $r = 3$) of the lost block.

$$SAD_{min} = \min_{n \in W} \sum_{\substack{1 \leqslant i \leqslant l \\ 1 \leqslant j \leqslant r}} |Dt_{i,j} - Ct_{i,j}|_n + |Db_{i,j} - Cb_{i,j}|_n \tag{4.24}$$

If $SAD_{min}$ is greater than a decision threshold related to the number of samples $n_{pel}$ used in the computation, i.e., $k \times SAD > n_{pel}$ then the lost depth block is not recovered with disparity compensation, but is further processed in the following stages. The constant $k = 20$ was found through an experimental study testing six sequences, as in Section 4.4.3 using several error patterns. The amount of processing and concealment done at *Stage 1* depends on $k$, but its actual value was found not to be a critical factor in the overall concealment performance.

### Depth correction

Although depth maps from adjacent views are highly correlated, there might be slight differences between their average depth values for the same region of the 3D scene. This is equivalent to having different average intensities between depth maps *N* and *N-1*. Thus, intensity correction is performed after recovering a lost depth block, which requires edge information to distinguish regions with different intensity. In the proposed method, the *Canny* edge detection algorithm is used for such purpose. Figure 4.17 illustrates an example of the underlying process devised for intensity correction using depth map edges. In this example, a depth Candidate Block from *view N-1* was used to recover a Lost Block in *view N*. The intensity correction is achieved by adding a correction factor $\Delta_{(i,j)}$ to each depth value $P_{(i,j)}$, as given by Equation 4.25,

$$P_{(i,j)} = P_{(i,j)} + \Delta_{(i,j)} \tag{4.25}$$

with $\Delta_{(i,j)}$ defined by Equation 4.26:

$$\Delta_{(i,j)} = \frac{(D_{(i,0)} - C_{(i,0)}).d2 + (D_{(i,h+1)} - C_{(i,h+1)}).d1}{d1 + d2}, \tag{4.26}$$

where $i$ and $j$ are the coordinates of depth value $P_{(i,j)}$ to be corrected, $h$ is the height of the block, $d1$ and $d2$ are the distances between $P_{(i,j)}$ and the top or bottom values in the concealed depth map. $D_{(i,0)}$ and $Db_{(i,h+1)}$, $C_{(i,0)}$ and $C_{(i,h+1)}$ are the top/bottom neighbour values in the depth map of *view N* and *view N-1*, respectively.

In the example illustrated in Figure 4.17, the interpolated value $P_{(i,j)}$ is located below the contour. In this case, the upper neighbour value $D_{(i,0)}$ is not used because it belongs to a region with different intensity, as defined by the contour. Thus, the difference between $D_{(i,0)}$ and $C_{(i,0)}$ is not considered and $d2 = 0$.

## 4.4.2 Stage 2: Depth contour reconstruction

In this stage, the same method as described in Section 4.3 is used, where the uncorrupted contour information from the texture image and the corrupted depth map are used in the concealment process.

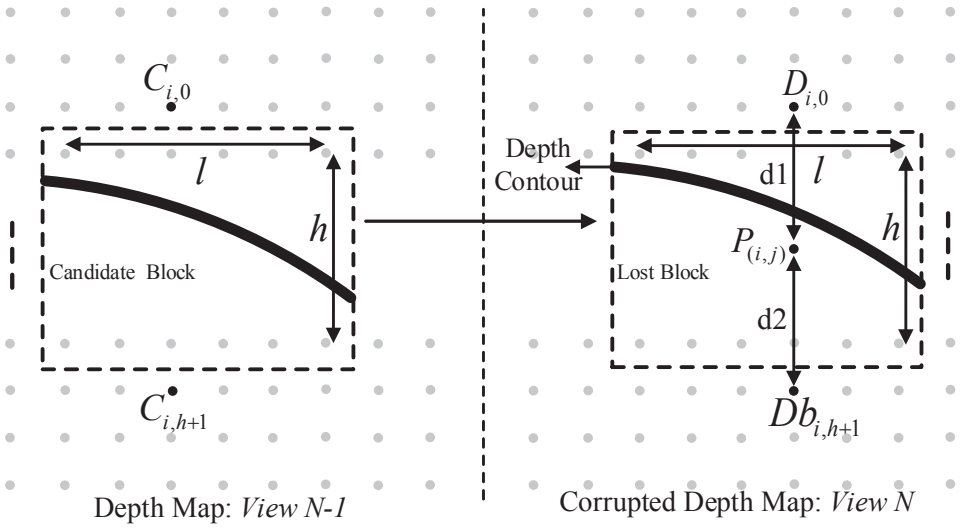



**Figure 4.17** – Intensity compensation of the recovered depth map.

### 4.4.3 Experimental results

The performance of the spatial EC algorithm presented in the previous subsections was evaluated using the MVD format with corrupted depth maps. The quality of the synthesised views (PSNR) was used as the performance metric of different spatial EC methods applied to depth maps used in the synthesis. The objective quality is evaluated using the virtual view synthesised with the uncorrupted depth map instead of the current view, to avoid the influence of the DIBR algorithm in the results.

The synthesis of virtual views was performed by using the reference software VSRS 3.5 (View Synthesis Reference Software) [29]. To obtain the necessary disparity map for *Stage 1*, the *OpenCV 2.1* software package was used [177, 178]. The contours of both the texture images and depth maps are extracted using the *Canny* edge detector [112] using the same implementation of Section 4.2.2.

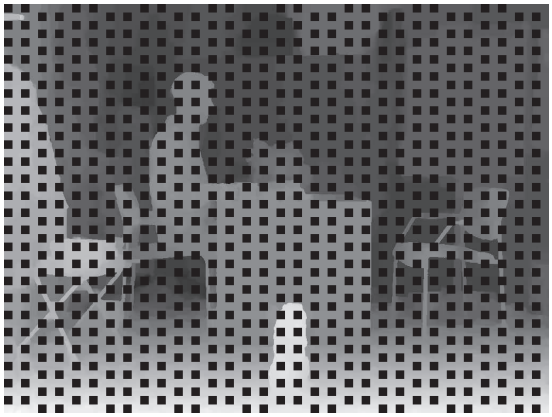
To evaluate and compare the performance of the proposed algorithm, two reference methods were used for spatial concealment of the same corrupted depth maps: (i) method R1 based on weighted spatial interpolation to fill in the missing depth regions using the uncorrupted depth values located along the borders of each missing region, the same reference method as used in Sections 4.1 and 4.2; (ii) method R2, based on disparity compensation using depth values of the uncorrupted adjacent view. No contour information

is used in reference methods R1 and R2. In the presence of error bursts affecting several consecutive blocks, weighted interpolation only uses the nearest (e.g., top and bottom) error-free border values ($N = 2$).

Two texture views and the corresponding depth maps of six sequences with 100 frames each were used in the simulations: *Book Arrival* (1024x768), *Balloons*, *Dancer* (1920x1080), *Shark* (1920x1080), *Kendo* (1024x768) and *Champagne* (1280x960). The visual content of each sequence has different characteristics, spatial resolution and different type of objects in the 3D scene, which results in a diverse dataset to evaluate the proposed algorithm. In sequence *Book Arrival view 8* and *view 10* were used to synthesise *view 9*, in *Balloons* and *Dancer*, *view 1* and *view 3* were used to synthesise *view 2*, in *Shark* and *Kendo view 1* and *view 5* were used to synthesise *view 3*, and in *Champagne view 37* and *view 39* were used to synthesise *view 38*.

To evaluate the performance of the proposed spatial EC method, the texture images and the corresponding depth maps used in the simulations were encoded with H.264/AVC Intra mode at fixed QP=30, using the reference software *JM17* [173]. Eight slice groups per depth map were packetized into one single packet, which never exceeded 1500 bytes in these tests. Two different types of slices were defined in slice mode and another one using the Flexible Macroblock Order (FMO) [115]. In the former, each slice comprises either one or two rows of macroblocks, while in the latter the dispersed mode was used. Note that FMO is not defined in the standard extensions based on the high profile, thus this case finds application in simulcast encoding schemes using the baseline profile. These packetization modes results in three different types of error patterns, as shown in Figure 4.18: (a) single macroblocks of $16 \times 16$ (b) rows of height=16; (c) rows of height=32. Packet losses were simulated using the Gilbert-Elliot model for different average packet loss ratios (PLR) and burst sizes [180]. Average PLRs of 5%, 10%, 20% and 40% with an average burst size of 3 were obtained from each sequence.

The proposed algorithm was evaluated in two different modes, referred to as PM(a) and PM(b) in Table 4.5. Mode PM(a) corresponds to the full algorithm, operating as depicted in Figure 4.15, while mode PM(b) is intended to evaluate the case where texture information is not available for any reason (e.g., lost, recovered from loss with poor quality). In the case of PM(b), the algorithm in Figure 4.15 follows from the start directly to Stage 3 and EC is fully obtained by using uncorrupted data from the depth map itself and Bézier



(a) Pack. mode 1

(b) Pack. mode 2

(c) Pack. mode 3

**Figure 4.18** – Error patterns originated from different packetisation modes.

curves, as described in Section 4.1. Note that this is the most difficult concealment case, because data from adjacent views is not used. The PSNR of the synthesised virtual views obtained by using depth maps, recovered with the proposed method, operating in modes PM(a) and PM(b), is compared with that obtained from reference methods R1 and R2, for different PLR and three types of packetization modes, previously described. Table 4.5 shows the difference ($\Delta_{PSNR}$) between the PSNR of synthesised views using the error-free depth map ($PLR = 0\%$ in column 0%) and the PSNR obtained from the various recovered depth maps ($PLR = x\%$). The lower this difference, the better the quality.

The case of full loss of the depth map was also simulated and the results are shown in Table 4.6. In this case, the proposed method (PM(c) in Table 4.6) uses the disparity compensated depth map of the adjacent view to substitute the entire lost depth map.

In the algorithm of Figure 4.15, this corresponds to use only *Stage 1*. The results are compared with those obtained from a copied depth map from the adjacent view (R3), i.e., the least complex method.

### Discussion

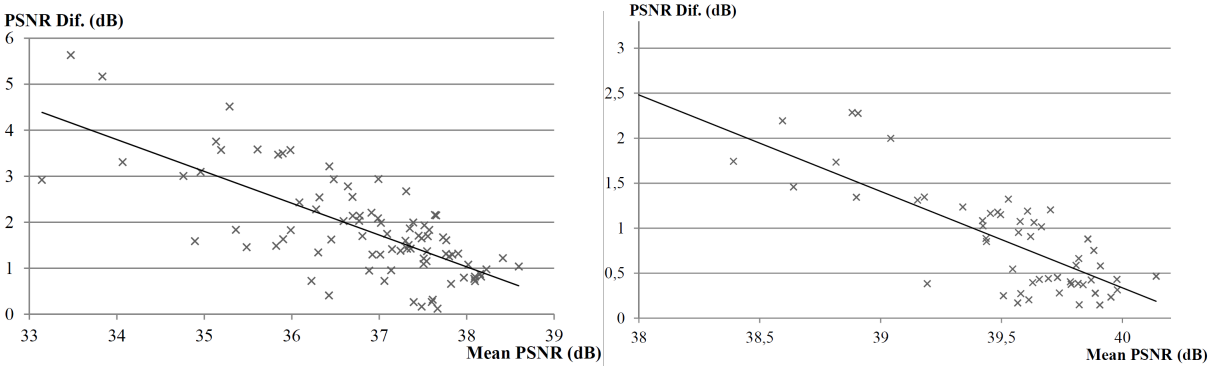
The experimental results shown in Table 4.5 demonstrate that the proposed method is able to consistently outperform the reference ones. The performance is better in sequences that have depth maps with many objects, because the lost areas are recovered with good accuracy, preserving the depth edges which are of major importance to achieve high quality synthesised images. The PSNR obtained by the proposed method is higher than the one achieved by the reference method R1, up to 4.04dB (*Champagne*), in the worst case scenario at 40% error loss in a depth map, with lost blocks of $16 \times 16$ (*packetisation mode 1*). The higher quality gain in Champagne is due to its higher spatial detail (i.e., more objects at different depth levels) in comparison with Balloons, for instance. Since Balloons is smoother than Champagne, which is a friendly characteristic for the reference method R1, the difference between the proposed and reference methods is smaller for Balloons. In general, the quality gain obtained with the proposed method is higher when recovering highly detailed missing regions, which are in fact the most difficult ones to reconstruct. This is the reason why the Champagne sequence provides better results than other sequences.

The lowest gains over the reference method R1 occur for slice modes where packet loss leads to missing regions with small height, such as, for instance *Pack. Mode 1*. In this case, particularly in depth maps with highly homogeneous regions, the results obtained from the reference methods are close to the proposed one (e.g., Dancer, Kendo). The advantage of using the proposed method increases at higher error rates with larger lost areas, as shown by the results in Table 4.5. This is due to the use of depth contours and selective spatial interpolation, which increase the reconstruction accuracy of depth values. The lowest gains over reference method R2 occur when the computed disparity is more accurate. In the synthetic sequences *Dancer* and *Shark*, and also on the natural video sequences *Book Arrival* and *Kendo*, the computed disparity map is more accurate than in sequences *Balloons* and Champagne, which leads to better results than R1, but still bellow the results of the proposed method PM(a). The proposed method PM(a) is also able to refine the disparity information with good accuracy, resulting in a good quality

recovered depth map. The disparity refinement process described in Section 4.4.1 is of major importance, especially in large lost areas, where the proposed method PM(b) and the reference methods cannot accurately conceal the depth map errors.

Table 4.6 shows that the proposed method consistently outperforms the reference one, as expected in this case, since copying the depth map from the adjacent view is a simple solution, but does not compensate for different view points.

To show the statistical behaviour of the proposed method in comparison with a reference (R1), a *Bland-Altman* plot is shown in Figure 4.19(a) and 4.19(b), corresponding to sequences Champagne e Book Arrival with losses of 20% and 10%, respectively. This plot shows that the difference between the proposed method and the reference one is larger for lower average PSNR, which demonstrates that its efficiency increases for higher loss rates, usually related to higher distortion. This behaviour is consistent for all sequences.




(a) Champagne (10% PLR) (b) Book Arrival (20% PLR)

**Figure 4.19** – Bland-Altman plots: proposed (PM(a)) vs. reference (R1), *Packetisation mode 1.*

The objective results can be subjectively confirmed in Figure 4.20 and 4.21, where the synthesised images with the respective depth maps are shown . The example of Figure 4.20 shows a region detail of the original depth map (Figure 4.20(a)), a corrupted depth map with two lost packets (Figure 4.20(b)), a depth map recovered with the reference method (4.20(c)), and a depth map recovered with the proposed method (Figure 4.20(d)). The bottom images of Figure 4.20 shows the corresponding synthesised views using the original depth map (Figure 4.20(e)), a depth map recovered with the reference method R2 (Figure 4.20(f)), and the synthesised views using the proposed method (Figure 4.20(g)). These details clearly show the type of artefacts caused by synthesis with inaccurate depth.

The highlighted regions in Figure 4.20 and Figure 4.21 show the impact of depth map accuracy in view synthesis. The detailed regions show that even when large depth areas are lost, the proposed method is able to reconstruct the lost regions with very good quality. When overlapped and thin objects coexists in the image, like the chair in Figure 4.21, the proposed method presents a good performance because it is able to restore the broken contours with higher fidelity using information from the corresponding texture image.

| | | | Pack. mode 1 | | | | Pack. mode 2 | | | | Pack. mode 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\Delta_{PSNR} = PSNR_{0\%} - PSNR_{x\%}$ | | | | | | | | | | | |
| Sequence | Method | 0% | 5% | 10% | 20% | 40% | 5% | 10% | 20% | 40% | 5% | 10% | 20% | 40% |
| Book Arrival | PM(a) | 40.27 | 0.48 | 0.59 | 0.75 | 1.70 | 0.23 | 0.33 | 0.33 | 0.72 | 0.32 | 0.35 | 0.44 | 0.94 |
| | PM(b) | | 0.44 | 0.54 | 0.68 | 1.73 | 0.63 | 0.81 | 1.14 | 2.81 | 0.68 | 0.80 | 1.09 | 2.36 |
| | R1 | | 0.65 | 0.92 | 1.30 | 3.67 | 0.68 | 0.82 | 1.05 | 2.59 | 0.99 | 1.23 | 1.55 | 3.40 |
| | R2 | | 2.31 | 2.56 | 3.16 | 5.04 | 0.60 | 0.81 | 0.89 | 1.53 | 0.63 | 0.73 | 0.88 | 1.57 |
| Balloons | PM(a) | 41.53 | 0.64 | 0.73 | 0.96 | 2.34 | 0.57 | 0.64 | 0.82 | 1.75 | 0.80 | 0.76 | 1.14 | 2.17 |
| | PM(b) | | 0.61 | 0.69 | 0.95 | 2.47 | 0.56 | 0.63 | 0.85 | 1.95 | 0.95 | 0.97 | 1.50 | 3.04 |
| | R1 | | 0.87 | 1.21 | 1.73 | 5.13 | 0.76 | 0.90 | 1.18 | 2.95 | 1.24 | 1.37 | 1.96 | 4.27 |
| | R2 | | 1.40 | 1.47 | 1.87 | 3.11 | 1.77 | 2.03 | 2.42 | 3.90 | 1.96 | 1.91 | 2.47 | 3.97 |
| Dancer | PM(a) | 38.04 | 0.03 | 0.12 | 0.19 | 0.74 | 0.03 | 0.08 | 0.08 | 0.16 | 0.02 | 0.09 | 0.10 | 0.21 |
| | PM(b) | | 0.16 | 0.32 | 0.38 | 0.92 | 0.24 | 0.37 | 0.42 | 1.19 | 0.30 | 0.56 | 0.65 | 1.80 |
| | R1 | | 0.34 | 0.65 | 0.95 | 3.28 | 0.11 | 0.21 | 0.24 | 0.85 | 0.17 | 0.35 | 0.38 | 1.26 |
| | R2 | | 0.48 | 0.69 | 0.83 | 1.37 | 0.57 | 0.71 | 0.86 | 1.49 | 0.55 | 0.73 | 0.86 | 1.49 |
| Shark | PM(a) | 39.77 | 0.33 | 0.38 | 0.37 | 0.78 | 0.26 | 0.29 | 0.20 | 0.49 | 0.25 | 0.28 | 0.21 | 0.21 |
| | PM(b) | | 0.48 | 0.58 | 0.63 | 1.31 | 0.46 | 0.58 | 0.64 | 1.45 | 0.71 | 0.83 | 1.04 | 2.37 |
| | R1 | | 0.58 | 0.76 | 0.94 | 2.34 | 0.59 | 0.74 | 0.83 | 1.88 | 0.83 | 0.96 | 1.16 | 1.26 |
| | R2 | | 0.62 | 0.67 | 0.71 | 1.23 | 0.64 | 0.73 | 0.75 | 1.44 | 0.65 | 0.71 | 0.78 | 1.49 |
| Kendo | PM(a) | 41.79 | 1.00 | 1.15 | 1.64 | 3.25 | 1.10 | 1.15 | 1.51 | 2.40 | 1.02 | 1.25 | 1.63 | 2.74 |
| | PM(b) | | 0.82 | 0.92 | 1.58 | 3.44 | 1.03 | 1.10 | 1.44 | 2.42 | 1.07 | 1.29 | 1.77 | 3.30 |
| | R1 | | 1.24 | 1.59 | 2.25 | 5.82 | 1.15 | 1.15 | 1.60 | 3.09 | 1.37 | 1.64 | 2.23 | 4.07 |
| | R2 | | 1.70 | 1.99 | 2.41 | 3.70 | 1.76 | 1.94 | 2.46 | 3.69 | 1.66 | 1.96 | 2.49 | 3.69 |
| Champagne | PM(a) | 38.95 | 0.80 | 1.28 | 1.58 | 3.30 | 0.70 | 0.92 | 1.32 | 2.72 | 0.73 | 0.86 | 1.20 | 2.40 |
| | PM(b) | | 0.94 | 1.43 | 1.75 | 3.40 | 1.91 | 1.91 | 3.89 | 9.83 | 2.40 | 3.43 | 4.88 | 9.98 |
| | R1 | | 1.46 | 2.27 | 2.94 | 6.14 | 1.17 | 1.52 | 1.91 | 4.34 | 2.02 | 2.37 | 3.05 | 5.55 |
| | R2 | | 3.44 | 3.97 | 4.40 | 7.37 | 3.80 | 4.21 | 4.94 | 8.54 | 3.72 | 4.10 | 5.00 | 7.06 |

**Table 4.5** – PSNR (dB) of synthesised views using the recovered depth maps under three different packetization modes and four PLR.

| | | | $\Delta_{PSNR} = PSNR_{0\%} - PSNR_{x\%}$ | | | |
|---|---|---|---|---|---|---|
| Sequence | Method | 0% | 5% | 10% | 20% | 40% |
| Book Arrival | PM(c) | 40.27 | 2.59 | 2.69 | 2.64 | 2.68 |
| | R3 | | 8.10 | 8.40 | 8.21 | 8.45 |
| Balloons | PM(c) | 41.53 | 6.48 | 6.46 | 6.56 | 6.41 |
| | R3 | | 9.39 | 9.33 | 9.41 | 9.28 |
| Dancer | PM(c) | 38.04 | 2.35 | 3.07 | 2.87 | 2.96 |
| | R3 | | 2.53 | 3.37 | 3.29 | 3.38 |
| Shark | PM(c) | 39.77 | 2.06 | 3.07 | 2.09 | 2.85 |
| | R3 | | 2.73 | 3.62 | 3.46 | 3.47 |
| Kendo | PM(c) | 41.79 | 6.03 | 6.37 | 6.06 | 6.17 |
| | R3 | | 9.41 | 9.39 | 9.41 | 9.38 |
| Champagne | PM(c) | 38.95 | 7.92 | 8.20 | 8.18 | 8.39 |
| | R3 | | 11.76 | 11.83 | 11.87 | 11.86 |

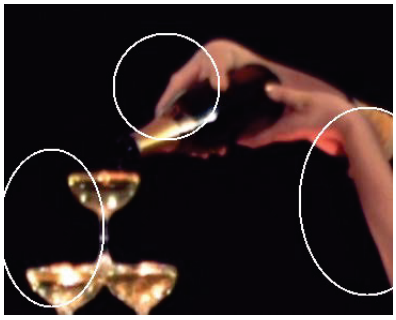
**Table 4.6** – PSNR (dB) of synthesised views using the recovered depth maps in the case of full frame loss.

(a) Original

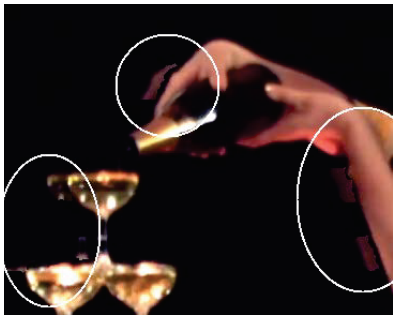
(b) Corrupted

(c) Reference method (R2)

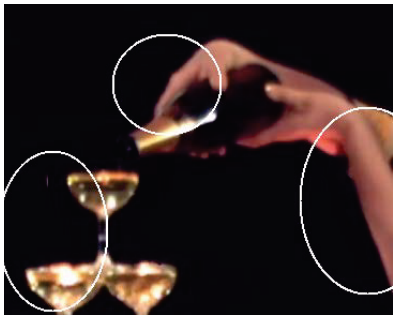
(d) Proposed method (PM(a))



(e) Original



(f) Reference method (R2) for an average video PSNR=33.95dB

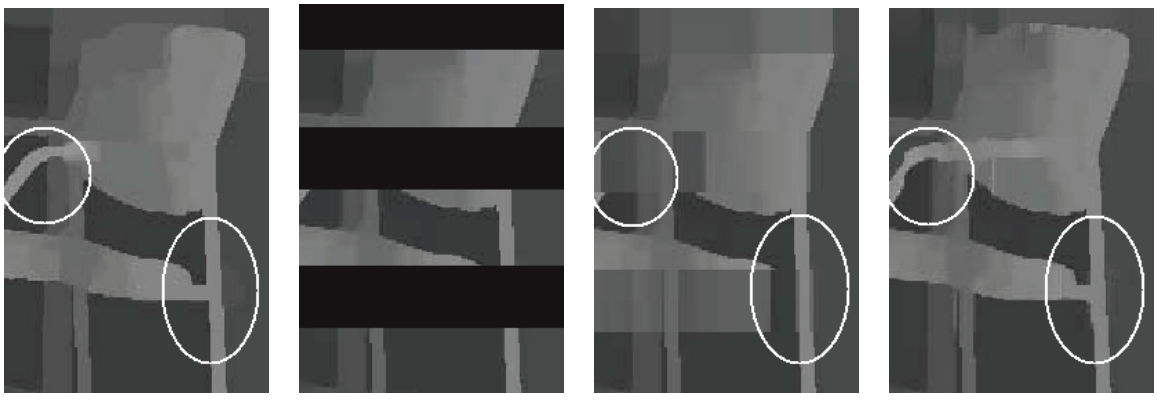


(g) Proposed method (PM(a)) for an average video PSNR=37.75dB

**Figure 4.20** – Example 1: Depth maps and respective synthesised views (example of 2 lost packets using packetization 3 in the 1st frame of *Champagne*), at 20% PLR.

When a small number of objects are present in the scene there are fewer variations of depth values in the depth map and, consequently, the depth map is more homogeneous, with less corresponding contours. In this case it is much simpler to recover a depth map and the performance of the proposed method is, in some cases, identical to the reference methods. When large variations of depth values are present in the depth map, the results show that the proposed method significantly outperforms the reference ones. By observing

the details of the synthesised view, particularly in regions with abrupt depth changes, one can confirm that the proposed method is able to achieve a better reconstruction than the reference.



(a) Original (b) Corrupted (c) Reference method (R1) (d) Proposed method (PM(a))

(e) Original (f) Reference method (R1) for an average video PSNR=39.18dB (g) Proposed method (PM(a)) for an average video PSNR=39.83dB

**Figure 4.21** – Example 2: Depth maps and respective synthesised views (example of 2 lost packets using packetization 3 in the 1st frame of *Book Arrival*), at 20% PLR.

## 4.5 Depth map error concealment for using BMGT

The EC method described in this section is tailored for multiview images-plus-depth, where at least two texture images exist along with the associated depth maps. In this work it is assumed that only the depth map of one view is affected by errors resulting in missing blocks/regions. This method is structured in three stages, as shown in Figure 4.23. As in previous methods, the first step is to extract the contours of the damaged depth map using the *Canny* edge algorithm [112]. For each lost block to be recovered, the homogeneity of the surrounding area is checked, based on the presence of edges. If the region around the block is classified as homogeneous, then the lost depth values are interpolated using a weighted average of the undamaged neighbour values (*Stage 1*). If the lost block is non-homogeneous, *Stage 2* is performed. Finally, if the lost area is not fully recovered in *Stage 2*, then *Stage 3* is performed. The three stages are briefly described as follows:

**Stage 1:** Weighted interpolation of the lost values using the non-corrupted neighbour values only.

**Stage 2:** EC using warping vectors obtained from Block Matching using Geometric Transformations (BMGT) on the texture images. Figure 4.22 shows an example of how a lost depth region is recovered using BMGT. The warping vectors are found by searching in regions of both texture images, co-located with the lost region of the corrupted depth map. The warped quadrilateral mapping is then used to interpolate the lost region, by using values of the non-corrupted depth map (see Figure 4.22).

**Stage 3:** Weighted interpolation of the lost values using the non-corrupted neighbour values and the depth contours, which are reconstructed based on the edge information of the texture image. Details of this method can be found in our previous work [11] described in Section 4.2.

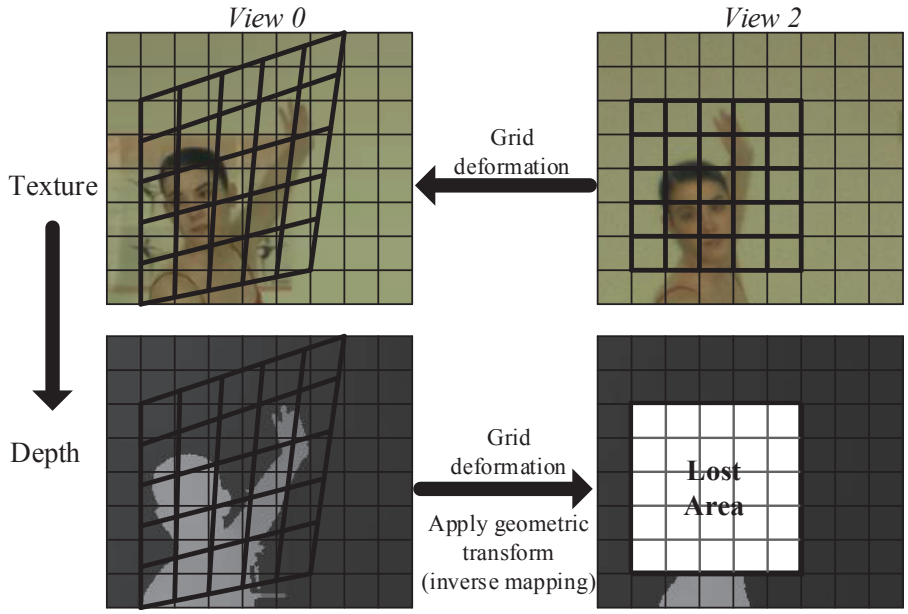



**Figure 4.22** – Depth error concealment with BMGT.

### 4.5.1 Block Matching with Geometric Transforms

A multiview-video system may use different camera arrangements, like linear or circular. In spite of this, and in the presence of complex motion in the scene, it is difficult to find a region match from another view/frame. Since complex motion and circular camera arrangements pose for more complex geometric distortions between adjacent frames, using a typical block matching technique (BMA) may not be efficient enough. In order to overcome this problem, an efficient region matching algorithm is exploited, which is capable of capturing complex motion between two frames [181]. This technique can also be used to represent complex motion between adjacent views, therefore it is proposed for the new depth map error concealment algorithm, because it allows recovering the lost depth areas more efficiently.

In a typical *BMA*, motion is represented by a simple translation of a rectangular block. In the *BMGT* algorithm, a block is deformed to achieve the best possible representation of a complex motion. The block deformation is performed through image warping, by means of a geometric mapping. Figure 4.24 represents an example of a quadrilateral that is warped from one view to another (Forward mapping). The parameters of a geometric transformation are computed by solving a group of equations relating the corners of a

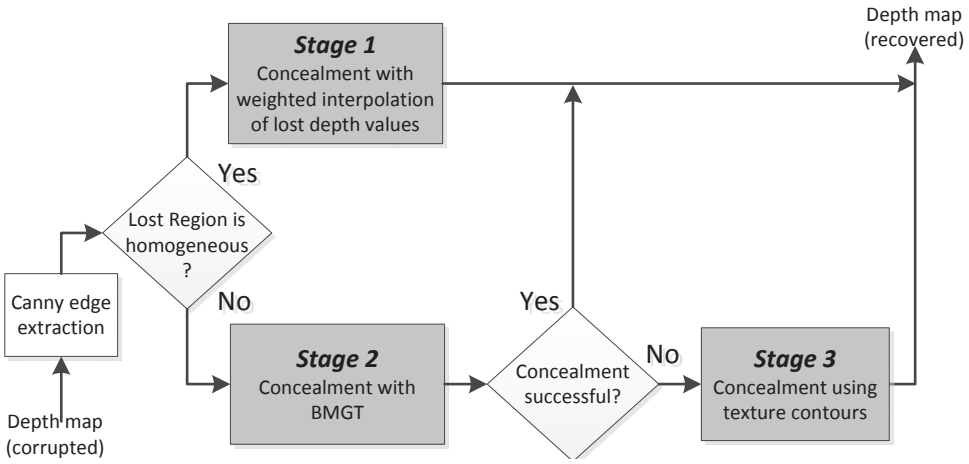



**Figure 4.23** – Proposed algorithm diagram with BMGT.

block (vertex points) $P_0$, $P_1$, $P_2$, $P_3$ from one quadrilateral to another. Each point of the quadrilateral $P_0...P_3$ is moved within a search window ($SW$) to minimise the distortion between values of the reference quadrilateral ($View\ N$) and the distorted quadrilateral in $View\ N-1$.

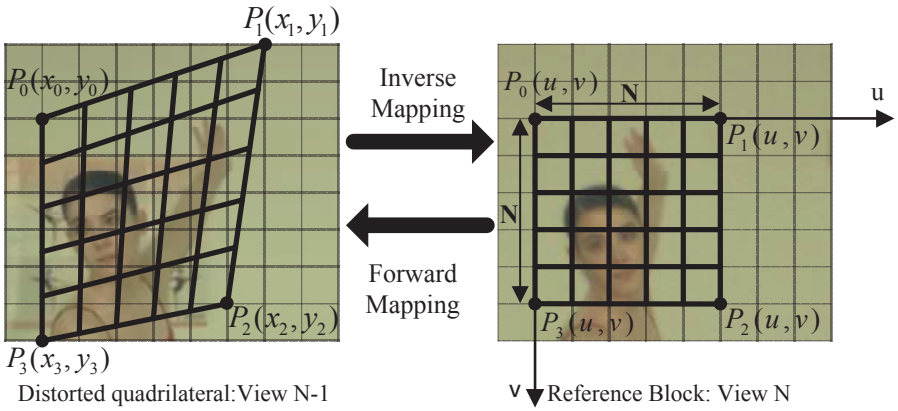



**Figure 4.24** – Example of a grid deformation used in BMGT.

Among the perspective transforms that may represent this type of geometric transformation applied to quadrilateral blocks, the *Bilinear* transformation has been chosen, not only due to its computational simplicity, but also due to its ability to represent the desired complex motion [181]. A *Bilinear* mapping is represented by Equation 4.27:

$$[x, y] = [uv, u, v, 1] \begin{bmatrix} a_3 & b_3 \\ a_2 & b_2 \\ a_1 & b_1 \\ a_0 & b_0 \end{bmatrix} \tag{4.27}$$

In order to determine the coordinates $X$ and $Y$ of a certain mapped values with bilinear interpolation Equations 4.28 and 4.29 are used:

$$X(u, v) = a_0 + a_1 v + a_2 u + a_3 uv \tag{4.28}$$

$$Y(u, v) = b_0 + b_1 v + b_2 u + b_3 uv \tag{4.29}$$

Considering Figure 4.24, where a $N \times N$ square block ($uv$ plane) is mapped onto an arbitrary quadrilateral ($xy$ plane), the corners are defined by the set of Equations 4.30:

$$\begin{aligned} (u, v) &= (x, y) \\ (0, 0) &= (x_0, y_0) \\ (N, 0) &= (x_1, y_1) \\ (N, N) &= (x_2, y_2) \\ (0, N) &= (x_3, y_3) \end{aligned} \tag{4.30}$$

Solving the previous equations in order to coefficients $a_k$, the following set of Equations 4.31 are determined, for $a_0$, $a_1$, $a_2$, $a_3$:

$$\begin{aligned} a_0 &= x_0 \\ a_1 &= \frac{x_3 - x_0}{N} \\ a_2 &= \frac{x_1 - x_0}{N} \\ a_3 &= \frac{x_2 - x_3 - x_1 + x_0}{N^2} \end{aligned} \tag{4.31}$$

Replacing the coefficients $a_k$ in Equations 4.28 and 4.29, the final equations that represent the coordinates $X$(Equation 4.32) and $Y$(Equation 4.33) of the mapped values are obtained :

$$X_{(u',v')} = x_0 + \frac{x_1 - x_0}{N}u' + \frac{x_3 - x_0}{N}v' + \frac{x_1 - x_3 - x_1 + x_0}{N^2}u'v' \tag{4.32}$$

$$Y_{(u',v')} = y_0 + \frac{y_1 - y_0}{N}u' + \frac{y_3 - y_0}{N}v' + \frac{y_1 - y_3 - y_1 + y_0}{N^2}u'v' \tag{4.33}$$

Since the coordinates $X$ and $Y$ are not integer values, it is not possible to obtain directly the mapped pixel or depth value, onto an exposed pixel grid, thus, it is necessary to interpolate the mapped values. In order to perform this operation, a bilinear interpolation is used based on the intensity of the four nearest neighbour values.

Figure 4.25 shows an example where a mapped value $P_{(u',v')}$ is interpolated using the four nearest values $P_a$, $P_b$, $P_c$ and $P_d$. $p_x$ and $p_y$ are, respectively, the horizontal and vertical distance between $P$ and $P_a$, where values $p_x$ and $p_y$ belong the interval $[0, 1]$.

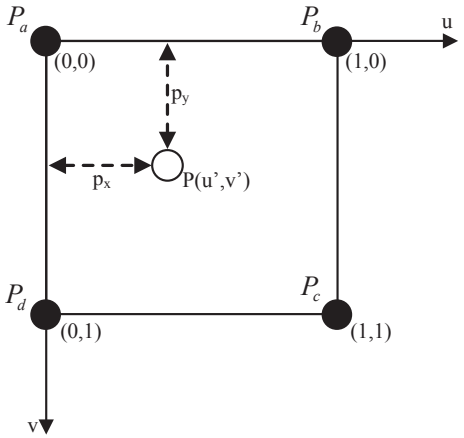



**Figure 4.25** – Grid Interpolation.

The point $P$ is defined by Equation 4.34:

$$P = (P_b - P_a)p_x + (P_d - P_a)p_y + (P_c - P_d - P_b + P_a)p_x p_y + P_a \tag{4.34}$$

After computing all values of the mapped quadrilateral, it is possible to measure the distortion between the mapped quadrilateral and the reference block. The Sum of Absolute Differences (SAD) was chosen as the distortion measure, given by Equation 4.35.

$$SAD = \sum_{y=0}^{B-1} \sum_{x=0}^{B-1} (|p_M(x,y) - p_r(x,y)|) \tag{4.35}$$

where $B$ is the block size, $p_M$ is the mapped quadrilateral and $p_r$ is the reference block.

### 4.5.2 Fast search with BMGT

The multiview video sequences used in this work present a disparity between two views that is lower than 64 pixels, thus in general the motion vectors norm is not higher than 64. For an object displacement between two images of $w$ pixels, the maximum search points (mappings) $p$ for each corner of the quadrilateral is given by Equation 4.36:

$$p = (2w + 1)^2 \tag{4.36}$$

Because the four vertices of the quadrilateral are displaced to all possible positions, the total number of search points ($M->$ mappings) is given by Equation 4.37

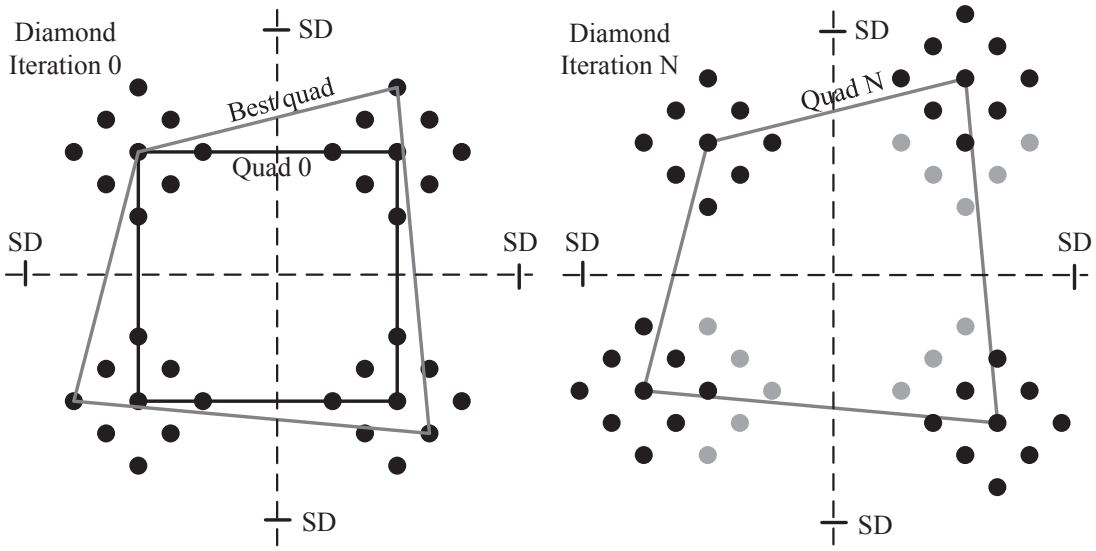
$$M = p^4 = (2w + 1)^8 \tag{4.37}$$

For $w = 64$, full search with BMGT may perform up to $7.67 \times 10^{16}$ quadrilateral mappings. For each mapping it is required to compute the SAD between the mapped values and the reference block (see Equation 4.35). Thus, using full search with BMGT to find a match for each combination is highly time consuming and not practical for an error concealment algorithm.

In order to minimize the computational complexity, a fast search algorithm was used, comprising two steps. Firstly a typical BMA with full search is used, typically comparing a rectangular block of pixels to another rectangular block of pixels. This initial search is done inside the limits of a search window ($SW$), allowing to find the translational motion component.

The second step is a refinement process, where the corners of the quadrilateral are displaced after the initial block matching, not relying on full search algorithm. The search method is based on the *Diamond* search, similar to the one implemented in the H.264/AVC

reference software [182]. In each corner of the quadrilateral, a diamond with nine points is positioned (see Figure 4.26), and then displaced to the diamond positions. During the first iteration (see Figure 4.26(a)), the nine-point diamond allows for $9^4$ mappings, covering a search window of six pixels. In the following iteration, the diamond grid is centred on the best position of the previous iteration (see Figure 4.26(b)). The search process stops whenever a search window boundary is reached ($SD$), or due to an early termination condition. If the SAD of a mapped quadrilateral is the same of the previous iteration the search is stopped.




(a) Grid Interpolation: iteration 0.

(b) Grid Interpolation: iteration N.

**Figure 4.26** – Quadrilateral diamond search.

### 4.5.3 Error concealment using BMGT

As mentioned before, when a lost block is located in a non-homogeneous region, *Stage 2* is executed using *BMGT*. The processing sequence in *Stage 2* is shown in Figure 4.27 and comprises the following steps:

**Step 1:** The texture image associated with the corrupted depth map ($View$ 2) is used as a reference to perform a *BMGT* search, in order to find the matching quadrilateral in the texture image of $View$ 0. The BMGT search is performed using the fast search technique

described in Section 4.5.2. In the translational block matching search, a window of 64 pixels ($SW = 64$) is used, and for the $BMGT$ search a 8 pixel ($SD = 8$) window is used.

**Step 2:** In this step, taking the best quadrilateral match for the texture image $View\ 0$ in the previous step, the depth map values of $View\ 0$ are used to recover the lost area of $View\ 2$ (inverse mapping). The matching refinement is performed by warping the quadrilateral area until the best possible match is found in the lost region, also using the method described in Section 4.5.2. The mapped depth values are candidates to fill the missing ones in the lost region. The match is verified by evaluating the distortion between depth values of the mapped candidate block (in $View\ 0$) and the non-corrupted depth values in the neighbourhood of the lost area (in $View\ 2$). Three rows (top and bottom neighbour values) and three columns (left and right neighbour values) of depth values are used to measure the distortion between the candidate block and the surrounding area of the region to be recovered. In case of error bursts only top and bottom neighbour values are used, as the left and right neighbours are unavailable. $SAD$ is computed to measure the distortion and a pre-defined threshold ($th$), defined as $th = 100 * N_{pel}$, is used to decide whether the mapped values from $View\ 0$ are suitable to recover the lost region of $View\ 2$. $N_{pel}$ is the number of depth neighbour values used to compute $SAD$ and the constant 100 was empirically obtained from several experiments. If the computed distortion (i.e., $SAD$) corresponding to the best quadrilateral used to recover the lost depth region is larger than $th$, this block is discarded and $BMGT$ is not used to recover the lost region.

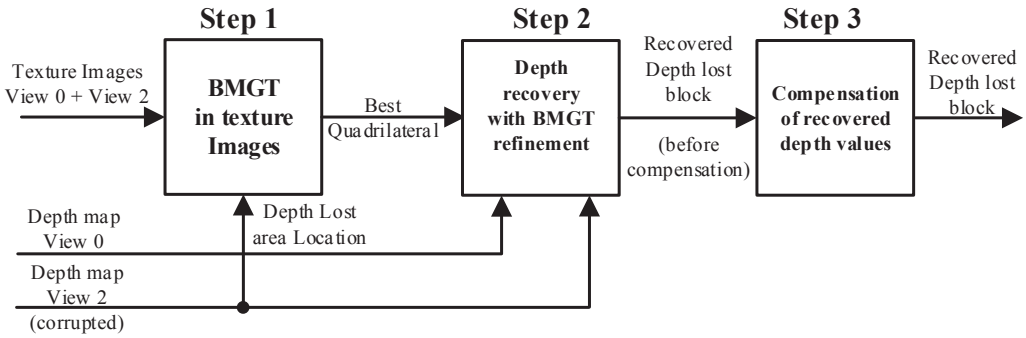



**Figure 4.27** – Stage 2: Functional Diagram.

**Step 3:** Since each view is obtained from a different camera position, the intensity of the depth map values used to recover the lost region from one view to another may be

different from the actual ones. Therefore, after recovering the lost depth values in *Step 2*, intensity compensation is performed by taking into account the edge information of the recovered block extracted by the *Canny* algorithm. The method to perform intensity compensation is the same as the one described in the Section 4.4.1.

### 4.5.4 Experimental Results

The performance of the proposed method was evaluated by using the first texture images and depth maps from two views (*Left View* and *Right View*) of six sequences, as described in Table 4.7. The reference software View Synthesis Reference Software (VSRS 3.5) [29], from the MPEG Group, was used to synthesise the second view. The contours of texture images and depth map were obtained with the *Canny* algorithm, as previously described.

**Table 4.7** – Used images corresponding with the respective cameras.

| Sequence | Resolution | Left View | Synth. View | Right View |
|---|---|---|---|---|
| Ballet | 1024×768 | camera 0 | camera 1 | camera 2 |
| Balloons | 1024×768 | camera 1 | camera 2 | camera 3 |
| Book Arrival | 1024×768 | camera 8 | camera 9 | camera 10 |
| Breakdancers | 1024×768 | camera 0 | camera 1 | camera 2 |
| Champagne | 1280×960 | camera 39 | camera 40 | camera 41 |
| Dancer | 1920×1080 | camera 1 | camera 2 | camera 3 |

The texture images and the corresponding depth maps are encoded with *H.264/AVC* using the same encoding configuration as the other methods described in previous sections of this chapter. To simulate errors in the depth maps, two types of error patterns were defined: single blocks of size $64 \times 64$, equally spaced throughout the depth map (Error pattern 1); rows with height of 64 depth samples, where the width of the rows corresponds to the full horizontal resolution of the image under test (Error pattern 2). This corresponds to lose regions in the depth map with data loss rates of 5%, 10%, 20% and 40%. This type of error patterns may occur when one or more slices are lost in coded images using Flexible Macroblock Order (FMO) [115]. Using this type of encoding scheme, it is very likely that the majority of the lost blocks are scattered in a similar way after decoding the corrupted depth map.

The algorithm was evaluated by computing objective quality (PSNR) of the synthesised views, obtained from the original depth map, in comparison with the one obtained from the recovered depth map. Using different error rates and patterns, from 0% up to 40%,

these results are presented in Table 4.8. The column *(Proposed-Reference)* is the difference between the proposed and the reference methods.

The results show that the proposed method is able to outperform the reference one (weighted spatial interpolation), especially in sequences where the depth maps contain many different depth levels, corresponding to several objects in the scene at different distances from the viewer. The proposed method shows good efficiency to recover the lost regions, by preserving the objects geometry and depth edges, which are of major importance to achieve high quality synthesised images. The PSNR obtained with the proposed method is higher than the reference one, up to 5.61dB (e.g., *Champagne*), in the worst case scenario at 40% error loss rate with error pattern 1. The lowest gains over the reference method occur when the error pattern affects larger areas, i.e., *Error pattern 2*. In this case and when the depth maps are highly homogeneous, the reference method also achieves good results. This can be seen in the sequences *Balloons* and *Breakdancers*, because in these cases the depth maps are much more homogeneous than in the other sequences. The advantages of using the proposed method increases for larger error rates, as can be seen in Table 4.8.

**Table 4.8** – Scenario 1: Experimental image synthesis results.

| | | | Reference Method | | | | Proposed Method | | | | (Proposed - Reference) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Error percentage | 0% | 5% | 10% | 20% | 40% | 5% | 10% | 20% | 40% | 5% | 10% | 20% | 40% |
| | **Error pattern 1 (64×64 blocks)** | | | | | | | | | | | | | |
| PSNR (dB) | Ballet | 36.80 | 34.90 | 35.84 | 33.92 | 33.92 | 36.73 | 36.74 | 36.36 | 35.62 | 1.83 | 0.90 | 2.44 | 1.70 |
| | Balloons | 41.37 | 41.20 | 38.83 | 38.57 | 37.25 | 41.21 | 39.00 | 39.44 | 37.85 | 0.01 | 0.17 | 0.87 | 0.60 |
| | Book Arrival | 41.78 | 40.84 | 41.06 | 39.81 | 38.52 | 41.66 | 41.59 | 41.07 | 40.84 | 0.82 | 0.59 | 1.26 | 2.32 |
| | Breakdancers | 38.21 | 37.87 | 37.73 | 37.61 | 36.78 | 38.01 | 38.06 | 37.84 | 37.07 | 0.14 | 0.33 | 0.23 | 0.29 |
| | Champagne | 39.67 | 37.06 | 35.10 | 34.43 | 32.25 | 39.12 | 37.60 | 37.68 | 37.81 | 2.06 | 2.50 | 3.25 | 5.61 |
| | Dancer | 40.34 | 40.25 | 40.14 | 39.34 | 38.24 | 40.29 | 40.29 | 39.96 | 40.09 | 0.14 | 0.15 | 0.62 | 1.85 |
| | **Error pattern 2 (64×Row bursts)** | | | | | | | | | | | | | |
| PSNR (dB) | Ballet | 36.80 | 36.54 | 36.14 | 35.36 | 34.83 | 36.66 | 36.14 | 36.25 | 35.16 | 0.12 | 0.00 | 0.89 | 0.33 |
| | Balloons | 41.37 | 39.04 | 40.87 | 38.85 | 38.00 | 39.48 | 41.05 | 39.22 | 38.47 | 0.44 | 0.18 | 0.37 | 0.47 |
| | Book Arrival | 41.78 | 41.47 | 40.07 | 39.23 | 39.55 | 41.52 | 40.28 | 39.47 | 39.97 | 0.05 | 0.21 | 0.24 | 0.42 |
| | Breakdancers | 38.21 | 37.48 | 38.01 | 36.32 | 37.07 | 37.84 | 38.03 | 36.50 | 37.35 | 0.36 | 0.02 | 0.18 | 0.28 |
| | Champagne | 39.67 | 38.57 | 35.98 | 35.76 | 34.12 | 38.86 | 37.72 | 37.27 | 35.52 | 0.29 | 1.74 | 1.51 | 1.40 |
| | Dancer | 40.34 | 40.21 | 40.19 | 39.91 | 39.73 | 40.26 | 40.25 | 39.71 | 39.98 | 0.05 | 0.06 | -0.2 | 0.25 |

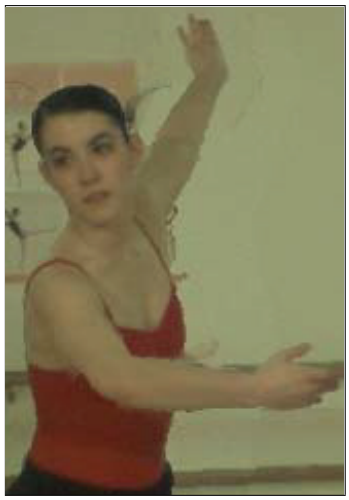
These results can be subjectively confirmed by observing Figure 4.28. This example shows details of: (a) the original depth map (Figure 4.28(a)), (b) the corrupted depth map with an error of 40% using *Error pattern 1* (Figure 4.28(b)), (c) depth map recovered with the reference method (Figure 4.28(c)), (d) depth map recovered with the proposed method (Figure 4.28(d)). Images at the bottom are the synthesised views obtained by using:

(a) Original.

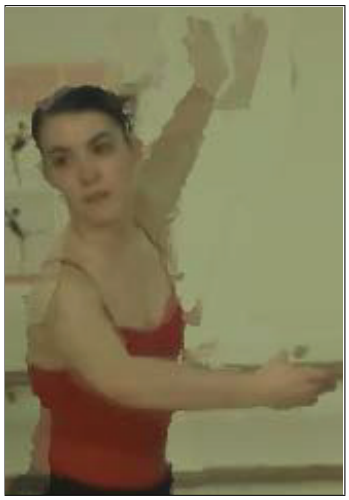
(b) Damaged.

(c) Ref. method.

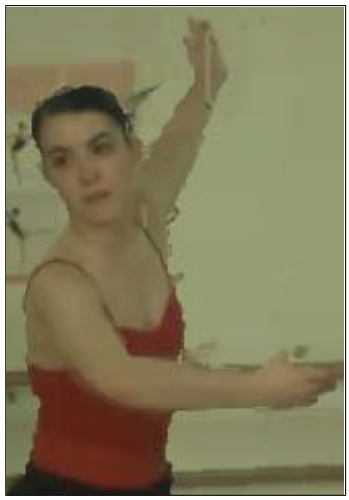
(d) Prop. method.



(e) Original.



(f) Ref. method.



(g) Prop. method.

**Figure 4.28** – Depth maps with the respective synthesised views (40% of block loss with error pattern 1).

(e) the original depth map (Figure 4.28(e)), (f) depth map recovered with the reference method (Figure 4.28(f)), and (g) depth map recovered with the proposed method (Figure 4.28(g)). These details clearly show the benefits of the proposed method.

Note that, in the example of Figure 4.28 the *ballerina* is rotated from one view to another. Even in the presence of such complex view geometry, the proposed method was able to accurately reconstruct the lost depth region. The advantage of using *Stage 2* in the proposed method is to complement the BMGT algorithm when it is not efficient, mostly when depth errors occur in the corresponding occluded areas between the two texture views.

### Discussion

In this chapter, five new error concealment methods for depth were proposed.

The technique described in Section 4.1 relies on the correctly decoded data on the corrupted depth itself to perform the EC. The technique described in Section 4.2, exploits texture images and depth maps similarities in order to recover corrupted depth map areas. The technique described in Section 4.3 combines the two previous methods, taking the advantages of both, resulting in an improved performance.

The method described in Section 4.4 exploits the similarities between texture images and depth maps from different views. In occluded regions this method as lower error concealment efficiency and in those cases, the technique proposed in Section 4.3 is used.

Despite the good efficiency of the technique described in 4.4, the proposed method described in 4.5 brought more flexibility, allowing to recover depth maps with good efficiency not only in sequences captured from parallel camera arrangements, but achieving good performance in more complex arrangements such as circular.

Depending on the available information in the concealment process, each one of those techniques has its particular benefits. All the proposed techniques have shown a good concealment performance, resulting in a set of EC techniques that are efficient in multiple types of scenarios.

# 5 Error concealment for multiview video-plus-depth

In this chapter, the work previously described is extended by exploiting temporal information in the concealment process, in order improve the accuracy of the reconstructed depth, whenever inter-frame coding parameter can be exploited. Geometric transforms are used to capture complex motion of depth sequences through the corresponding texture images. The EC method proposed in this chapter for depth maps is tailored for multiview video-plus-depth, where at least two texture video sequences are transmitted along with the associated depth maps. Additionally, the investigation is further extended to multipath transmission scenarios using multiple description coding (MDC) for depth maps. A method to recover lost descriptions in MDC is proposed to cope with the time-varying conditions of independent channels carrying different depth MDC streams.

## 5.1 Inter-view and inter-frame error concealment using BMGT

In this section an EC method for depth maps associated with multiview video is described, where the loss, of data may be due to transmission errors or packet loss, as in previous cases. The major novelty of the proposed method relies on exploiting the temporal information and similarity between adjacent texture views, using a block-matching algorithm based on geometric transforms.

In this work, it is assumed that only one of the depth sequences is possibly affected by losses, while others are received without error/losses. Extension to more than one sequence does not pose significant additional challenges. Figure 5.1 presents the general flowchart of the EC method developed in this section where the various processing stages are shown. Initially, the contours of the damaged depth map are extracted using an edge extraction algorithm, and then the coding mode, Inter or Intra, is detected for each depth region.

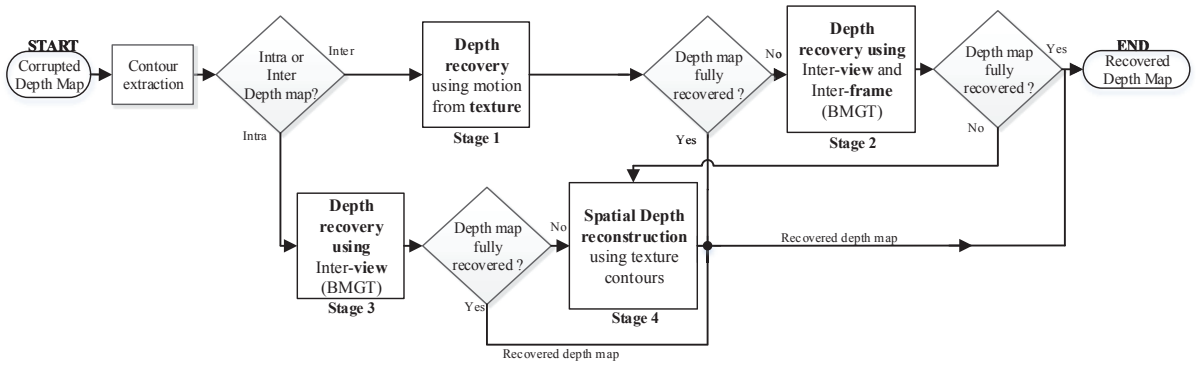



**Figure 5.1** – Block diagram of the proposed depth map error concealment method.

In case of inter-coded depth maps, motion compensation is performed in *Stage 1*, using motion and residual information from the corresponding texture image. If the depth map is not fully recovered after *Stage 1*, then the concealment process proceeds to *Stage 2*, where the remaining lost depth values are recovered using *Inter-view* and *Inter-frame* estimation techniques based on BGMT. Two candidates are determined for each lost value and the best candidate is chosen based on a boundary matching analysis. If the lost region of the depth map is still not fully recovered, a *Spatial Depth reconstruction* method is further performed in *Stage 4*.

In the case of Intra coded depth maps with lost slices, the concealment process begins at *Stage 3*, because only *inter-view* EC may be used, as for *inter-frame* motion information is not available. After *Stage 3*, if the depth map is not fully recovered, the remaining depth values are processed at *Stage 4*.

The proposed algorithm is able to recover isolated lost regions and full-frame depth map losses. When a full depth map is lost, *Spatial Depth recovery* cannot be performed, but *EC using residual and motion from texture* techniques is available (Section 5.1.1), as well

as *inter-view* (Section 5.1.2) and *inter-frame* (Section 5.1.2).

### 5.1.1 Error concealment using motion from texture

As shown in the algorithmic structure of Figure 5.1, when the depth map to be recovered is inter-coded, its recovery is done at *Stage 1* by using motion information from the corresponding texture image.

This is based on the evidence that texture motion information is highly correlated with that of depth maps [156]. Thus, this information is used to perform motion compensation in corrupted depth maps to reconstruct lost regions. However, not all motion information available in the texture is useful for depth motion compensated EC, because some motion vectors (MV) may not be accurate enough. The MV selection criterion is based on the texture residual information, where only texture MVs with null residual information are chosen. For any block $k$ in the texture image, only MVs $v_k$ are considered accurate enough. It is considered that for an block $k$, $r_{n,m}$ are the residue coefficients and $v_k^* \epsilon \{v_k : R_k = 0\}$, where $R_k = \sum_{n,m \epsilon k} |r_{n,m}|$. In this method, depth motion compensation does not use the less accurate MVs, thus error propagation is minimized in comparison with the case where all MVs are used, regardless the existence of residual information.

The remaining lost depth areas, that are not reconstructed due to the lack of accurate MVs, are concealed in following stages, as shown in Figure 5.1.

### 5.1.2 Depth error concealment using BMGT

#### Inter-view

The inter-view EC methods using BMGT are implemented in *Stage 2* and *Stage 4* of the algorithm presented in Figure 5.1. Since these techniques are based on the same approach as the method described in Section 4.5, no further details are given in this section.

### Inter-frame

The Inter-frame EC technique developed in this section is partially similar to one described in Section 5.1.2. The main difference relies on the depth information that is used to recover the lost regions. In this case, such information is retrieved from a previously decoded depth map of the same view, as the corrupted region. The motion information is gathered not only from the uncorrupted neighbouring regions of a temporally close depth map, but also from the texture image of the same view. Depending on the type of coded region to be recovered, either from a $P$ or $B$-frame, there are MVs pointing to past or future frames, previously decoded and stored in the decoder reference lists. The MVs obtained from the texture and depth are then selected as follows:

- Using MVs from texture regions co-located with the depth lost area, three different MVs are computed ($MV^T(i)$) as defined by Equation 5.1, where $V_x^T(i)$ and $V_y^T(i)$ are the horizontal and vertical components of $MV^T(i)$, respectively.

$$MV^T(i) = \left(V_x^T(i), V_y^T(i)\right), i \in \{min, max, mean\} \tag{5.1}$$

The index $i$ identifies $MV_i^T$ according to its magnitude, either the minimum, maximum or average of the MVs contained in region co-located with the lost area in the texture image.

- If the MVs of the neighbouring regions of the missing depth are available, then they are also taken into account to compute twelve different MVs ($MV^D(i,j)$), as defined by Equation 5.2. $V_x^D(i,j)$ and $V_y^D(i,j)$ are the horizontal and vertical components of $MV^D(i,j)$. $i \in \{min, max, mean\}$ correspond to the minimum, maximum and average magnitude, respectively, of the MVs located at the top, bottom, left and right of the missing depth region, as defined by $j \in \{Top, Bot, Left, Right\}$.

$$MV_{(i,j)}^D = (V_x^D(i,j), V_y^D(i,j)), i \in \{min, max, mean\}; j \in \{Top, Bot, Left, Right\} \tag{5.2}$$

Each of these 15 MVs (three computed from texture image and twelve from depth map) are used to define the start searching point of the quadrilateral for $BMGT$, which is

similar to the second step described in Section 5.1.2. Each MV will lead to a candidate block of depth values $Dcand_k$, where $Dcand_k$ has the same size of the missing block to be reconstructed ($k \in \{1, 2..., 15\}$).

After computing and storing all $Dcand_k$ in their respective buffers, the best $Dcand_k$ to reconstruct the lost region is chosen by minimising the respective SAD at the boundaries between the candidate block and the location of the region to be recovered. The minimum SAD value defines the best candidate $D^*cand_k$, but if the SAD is greater than a threshold ($th$), then the $Dcand_k$ block is discarded and $BMGT$ is not used to recover the lost region.

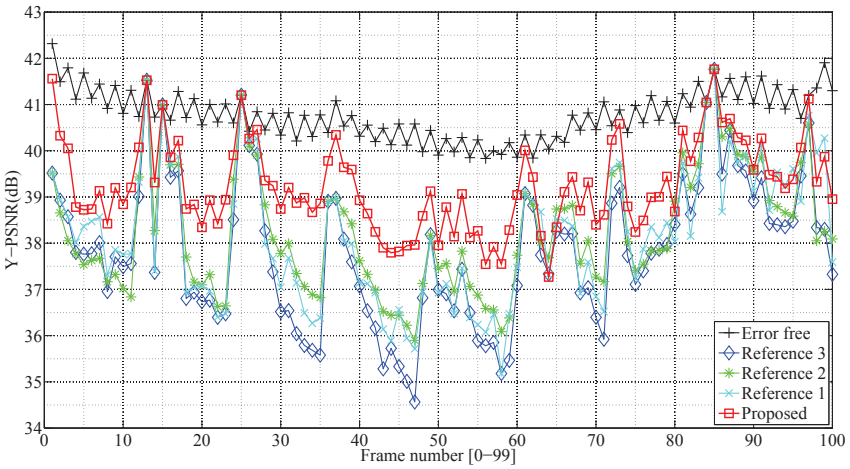
In the presence of large missing depth areas, due to error bursts or full frame losses, the neighbouring depth information is not considered reliable for refining the matching region. When more than 50% of the recovered neighbouring region is also lost, the *inter-frame* error concealment described in this subsection is not performed. In such cases, only *inter-view* EC is performed, as described in the beginning of this section.

### 5.1.3 Spatial recovery of missing depth

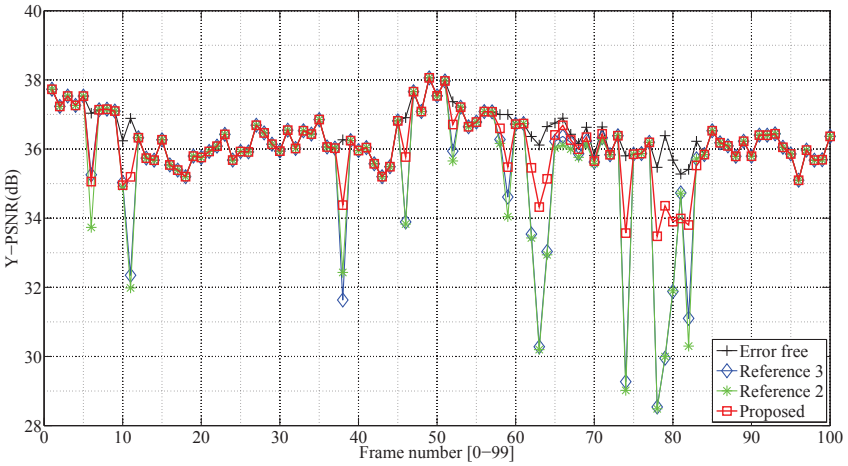
The remaining depth areas, i.e., those that were not recovered in *Stage 2* and *Stage 3*, are processed using a *Spatial Error Concealment* method implemented at *Stage 4*. In this method, lost depth values are interpolated using the non-corrupted neighbouring depth and recovered values in the previous stages. The depth contours are reconstructed based on the edge information of the texture image. Further details on this method can be found in Section 4.2.

**Table 5.1** – Tested video sequences.

| Sequence (no. frames) | Resolution | Left View | Synth. View | Right View |
|---|---|---|---|---|
| Ballet (100) | 1024×768 | camera 0 | camera 1 | camera 2 |
| Book Arrival (100) | 1024×768 | camera 8 | camera 9 | camera 10 |
| Breakdancers (100) | 1024×768 | camera 0 | camera 1 | camera 2 |
| Dancer (250) | 1920×1080 | camera 1 | camera 2 | camera 3 |
| Shark (300) | 1920×1088 | camera 1 | camera 3 | camera 5 |
| Champagne (300) | 1280×960 | camera 39 | camera 40 | camera 41 |




(a) *Shark* sequence, PLR=20% and Packetization 1.




(b) *Breakdancers* sequence, PLR=10% and Packetization 3.

**Figure 5.2** – Objective reconstruction quality (PSNR) of synthesised view luminances.

### 5.1.4 Simulation results and discussion

#### Experimental setup

The performance of the depth error concealment method described in the previous sections was evaluated using a diverse dataset of MVD sequences with two views and corresponding depth maps. Five sequences with different characteristics, such as spatial resolution and different types of objects in the 3D scene were used: *Ballet*, *Book Arrival*, *Breakdancers* and *Champagne* are natural scenes and sequences *Dancer* and *Shark* are digitally created.

The View Synthesis Reference Software (VSRS 3.5) [29] from the MPEG Group was used to synthesise virtual views using the various depth maps under evaluation. The *OpenCV 2.1* implementation of the Canny algorithm [177] was used to extract the contours of texture images and depth maps as in previous methods.

The texture images and the corresponding depth maps used in the simulations were independently encoded with H.264/AVC at fixed QP=28, using the reference software *JM17* [173]. The IDR period was set to 12 frames and the Group Of Pictures (GOP) structure is *IBPBP*, using 2 reference frames.

In order to evaluate the performance of the proposed algorithm a packet loss simulation environment was defined, using three packetization modes, which result in different error patterns. For *Packetization Mode 1*(PM-1), the FMO explicit mode was used with eight slices per frame, resulting in a dispersed checkerboard pattern with square blocks of $64 \times 64$ pixels. In *Packetization Mode 2*(PM-2), slice mode was used and each frame was divided into slices with height of 64 pixels by the whole width as the texture images. In *Packetization Mode 3* (PM-3), each depth frame corresponds to a single packet, resulting in full frame losses whenever a packet is lost.

Depending on the size of each coded slice, each packet may contain several slices for a packet size of up to 1500 bytes. In order to increase resiliency and reduce error bursts, interleaved packetization was used for slices in order to disperse the packet losses through the entire GOP. The following three reference methods were used for comparison.

**Ref. 1** This is a spatial error concealment method, based on bilinear interpolation, without taking into account any particular characteristics of depth maps, e.g., edges (See Section 3.2.1). In the case of entire frame loss, no spatial EC can be done, therefore this reference method is not used.

**Ref. 2** In this method, the texture MVs are used to recover the depth maps through motion compensation, similarly to the method implemented in [156].

**Ref. 3** In this reference method, the missing depth map regions or entirely lost depth maps are recovered using "*Frame copy*" method, as in H.264/AVC reference decoder *JM17*

[173].

### Results and discussion

The objective quality (average PSNR) of video sequences synthesised with depth maps recovered by the method previously described is compared with the quality obtained with depth maps recovered by the reference methods (Ref. 1, Ref. 2 and Ref. 3), as shown in Table 5.2. The difference between the PSNR obtained from the synthesised view using the error-free depth map (i.e., Column 0%) and that obtained from the synthesised view using the recovered depth maps is also shown for $\Delta Ref1$, $\Delta Ref2$, $\Delta Ref3$ and the proposed method ($\Delta Prop$). The lower this difference, the better the quality of depth map reconstruction. Using the PSNR differences, an objective comparison between the proposed method and the reference ones is also shown in columns: $\Delta Prop\_Ref1$, $\Delta Prop\_Ref2$, $\Delta Prop\_Ref3$. The higher this difference, the better is the proposed method when compared with the reference ones.

When large depth regions are lost, the reference methods tend to perform inefficiently, often resulting in poor error concealment. Depending on the spatial location of the errors, the negative effects of inaccurate recovery of missing depth can propagate throughout the GOP. These simulation results show that the proposed method is able to outperform the reference ones, especially in sequences where the depth maps contain many objects and large movements. This is because lost regions are recovered with higher accuracy, preserving the depth discontinuities, which are of major importance to achieve high quality in the synthesised images. In smooth areas, *Ref. 1* is able to recover the lost areas with high accuracy, as can be seen in the good results achieved for sequences *Dancer* and *Breakdancers*. Since *Ref. 2* uses MVs from the corresponding texture image, its performance is highly dependent on MV accuracy, because texture motion is not always highly correlated with depth motion. Both *Ref. 2* and *Ref. 3* show good performance in low motion sequences, because previously decoded frames provide perfect motion vector candidates for depth maps. For example, sequence *Champagne* presents low motion activity between frames and all methods not based on spatial EC perform quite well. *Ref. 2* is particularly efficient in sequence *Dancer*, where the depth maps are mainly comprised of smooth areas with a single moving object in the scene, i.e., the dancer. In most frames of the sequence, this dancer does not present a very complex motion, thus, the MVs extracted from the

**Table 5.2** – Mean PSNR-Y(dB) of synthesised views using recovered depth maps.

| | | ΔRef1 | | | | ΔRef2 | | | | ΔRef3 | | | | ΔProp | | | | ΔProp_Ref1 | | | | ΔProp_Ref2 | | | | ΔProp_Ref3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Err. / Seq. | 0% | 2% | 5% | 10% | 20% | 2% | 5% | 10% | 20% | 2% | 5% | 10% | 20% | 2% | 5% | 10% | 20% | 2% | 5% | 10% | 20% | 2% | 5% | 10% | 20% | 2% | 5% | 10% | 20% |
| | | **Packetization mode 1: Square Blocks** | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ballet | 36.14 | 0.88 | 1.55 | 3.05 | 4.89 | 0.39 | 0.66 | 1.52 | 2.34 | 0.41 | 0.73 | 1.68 | 2.54 | 0.13 | 0.28 | 0.67 | 1.48 | 0.75 | 1.27 | 2.38 | 3.41 | 0.26 | 0.38 | 0.85 | 0.86 | 0.28 | 0.45 | 1.01 | 1.06 |
| Book Arrival | 41.08 | 0.62 | 1.08 | 2.22 | 3.72 | 0.28 | 0.52 | 1.21 | 1.95 | 0.31 | 0.55 | 1.27 | 2.00 | 0.15 | 0.31 | 0.74 | 1.27 | 0.47 | 0.77 | 1.48 | 2.45 | 0.13 | 0.21 | 0.47 | 0.68 | 0.16 | 0.24 | 0.53 | 0.73 |
| Breakdancers | 36.35 | 0.20 | 0.43 | 0.95 | 1.78 | 0.26 | 0.56 | 1.16 | 2.07 | 0.28 | 0.60 | 1.21 | 2.22 | 0.16 | 0.33 | 0.68 | 1.21 | 0.04 | 0.10 | 0.27 | 0.57 | 0.10 | 0.23 | 0.48 | 0.86 | 0.12 | 0.27 | 0.53 | 1.01 |
| Dancer | 39.35 | 0.17 | 0.52 | 0.97 | 1.87 | 0.18 | 0.54 | 0.99 | 1.77 | 0.16 | 0.47 | 0.90 | 1.59 | 0.11 | 0.28 | 0.54 | 0.94 | 0.06 | 0.24 | 0.43 | 0.93 | 0.07 | 0.26 | 0.45 | 0.83 | 0.05 | 0.19 | 0.36 | 0.65 |
| Shark | 39.35 | 0.30 | 0.69 | 1.29 | 2.20 | 0.32 | 0.75 | 1.38 | 2.36 | 0.38 | 0.82 | 1.52 | 2.58 | 0.19 | 0.42 | 0.81 | 1.41 | 0.11 | 0.27 | 0.48 | 0.79 | 0.13 | 0.33 | 0.57 | 0.95 | 0.19 | 0.40 | 0.71 | 1.27 |
| Champagne | 38.77 | 1.34 | 2.84 | 4.87 | 6.97 | 0.47 | 0.99 | 1.82 | 2.79 | 0.49 | 1.02 | 1.86 | 2.85 | 0.25 | 0.56 | 1.11 | 1.94 | 1.09 | 2.28 | 3.76 | 5.03 | 0.22 | 0.43 | 0.71 | 0.85 | 0.24 | 0.46 | 0.75 | 0.91 |
| | | **Packetization mode 2: Rows** | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ballet | 36.15 | 1.64 | 2.83 | 4.87 | 5.05 | 0.55 | 1.31 | 1.86 | 2.44 | 0.42 | 1.15 | 1.86 | 1.93 | 0.39 | 0.77 | 1.11 | 1.64 | 1.25 | 2.06 | 3.76 | 3.41 | 0.96 | 0.54 | 0.75 | 0.80 | 0.03 | 0.38 | 0.75 | 0.29 |
| Book Arrival | 41.07 | 0.62 | 1.29 | 2.39 | 4.56 | 0.34 | 0.60 | 1.15 | 2.15 | 0.29 | 0.46 | 0.98 | 1.84 | 0.17 | 0.32 | 0.67 | 1.24 | 0.45 | 2.07 | 1.72 | 3.32 | 0.17 | 0.28 | 0.48 | 0.91 | 0.12 | 0.14 | 0.31 | 0.60 |
| Breakdancers | 36.34 | 0.23 | 0.63 | 1.27 | 2.74 | 0.36 | 0.75 | 1.62 | 2.82 | 0.37 | 0.70 | 1.59 | 2.79 | 0.13 | 0.42 | 0.95 | 1.31 | 0.10 | 0.21 | 0.32 | 1.43 | 0.23 | 0.33 | 0.67 | 1.51 | 0.24 | 0.37 | 0.64 | 1.48 |
| Dancer | 39.33 | 0.75 | 1.04 | 1.50 | 2.41 | 0.54 | 0.83 | 1.36 | 2.21 | 0.50 | 0.77 | 1.25 | 2.07 | 0.23 | 0.34 | 0.57 | 0.97 | 0.52 | 0.70 | 0.93 | 1.44 | 0.31 | 0.49 | 0.79 | 1.24 | 0.27 | 0.43 | 0.68 | 1.10 |
| Shark | 40.78 | 0.32 | 0.85 | 1.53 | 2.82 | 0.33 | 0.92 | 1.67 | 2.91 | 0.26 | 0.70 | 1.35 | 2.41 | 0.13 | 0.38 | 0.70 | 1.31 | 0.19 | 0.47 | 0.83 | 1.51 | 0.20 | 0.54 | 0.97 | 1.60 | 0.13 | 0.32 | 0.65 | 1.10 |
| Champagne | 38.70 | 1.08 | 3.02 | 4.60 | 7.06 | 0.18 | 1.01 | 1.51 | 2.49 | 0.17 | 1.00 | 1.48 | 2.42 | 0.15 | 0.52 | 1.08 | 2.07 | 0.93 | 2.50 | 3.52 | 4.99 | 0.03 | 0.49 | 0.43 | 0.42 | 0.02 | 0.48 | 0.40 | 0.35 |
| | | **Packetization mode 3: Full frame** | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ballet | 36.16 | - | - | - | - | 0.16 | 0.36 | 0.80 | 1.50 | 0.14 | 0.34 | 0.75 | 1.56 | 0.12 | 0.27 | 0.58 | 1.18 | - | - | - | - | 0.04 | 0.09 | 0.22 | 0.32 | 0.02 | 0.07 | 0.17 | 0.38 |
| Book Arrival | 41.16 | - | - | - | - | 0.16 | 0.33 | 0.81 | 1.51 | 0.13 | 0.30 | 0.71 | 1.48 | 0.11 | 0.28 | 0.60 | 1.22 | - | - | - | - | 0.05 | 0.05 | 0.21 | 0.29 | 0.02 | 0.02 | 0.11 | 0.26 |
| Breakdancers | 36.36 | - | - | - | - | 0.23 | 0.52 | 1.04 | 1.99 | 0.22 | 0.53 | 1.08 | 2.15 | 0.15 | 0.29 | 0.58 | 1.10 | - | - | - | - | 0.08 | 0.23 | 0.46 | 0.89 | 0.07 | 0.24 | 0.50 | 1.05 |
| Dancer | 39.29 | - | - | - | - | 0.13 | 0.27 | 0.78 | 2.01 | 0.10 | 0.20 | 0.60 | 1.68 | 0.06 | 0.11 | 0.32 | 0.91 | - | - | - | - | 0.07 | 0.16 | 0.46 | 1.09 | 0.04 | 0.09 | 0.28 | 0.77 |
| Shark | 40.78 | - | - | - | - | 0.23 | 0.73 | 1.54 | 2.51 | 0.18 | 0.59 | 1.22 | 2.19 | 0.12 | 0.18 | 0.42 | 0.67 | - | - | - | - | 0.11 | 0.55 | 1.12 | 1.84 | 0.06 | 0.41 | 0.80 | 1.52 |
| Champagne | 38.79 | - | - | - | - | 0.06 | 0.14 | 0.26 | 0.58 | 0.07 | 0.12 | 0.22 | 0.49 | 0.06 | 0.10 | 0.21 | 0.46 | - | - | - | - | = | 0.04 | 0.05 | 0.12 | 0.01 | 0.02 | 0.01 | 0.03 |

texture are quite accurate, which is an advantage for both, *Ref. 2* and the proposed method. This good performance is represented by small PSNR differences, not present in other sequences.

For most scenarios, 20% PLR using *packetization 2*, the quality of the synthesised views obtained by the proposed method is higher than the one achieved by the best reference method, up to 1.48dB (*Breakdancers*). This performance is consistent for different types of sequences and various reference methods. Some reference methods are also able to recover the depth maps with good accuracy, but when the characteristics of the image change (homogeneity of the depth, motion, etc.), both in the temporal and spatial domain, the quality of the recovered depth maps tend to decrease. In such conditions, the proposed method is still consistently better than its counterparts.

When comparing digitally created depth maps with others obtained from natural scenes, the latter are more prone to noise and imperfections. Depth maps computed for synthetic content usually have more accurate shapes than natural video, which results in sharper edges and a superior performance of the proposed method. This can be observed in the *Shark* sequence, which was digitally created and has a diverse and large number of well-defined shapes along the sequence.

These objective results can be subjectively confirmed in Figure 5.3, where some synthesised images and respective depth maps are shown. The example of Figure 5.3 shows a

region detail of the original depth map (Figure 5.3(a)), the corrupted depth map with a 10% PLR (Figure 5.3(b)), depth map recovered with the reference methods (Figure 5.3(c) to 5.3(e)), and a depth map recovered with the proposed method (Figure 5.3(f)). The bottom images of Figure 5.3 show the corresponding synthesised views using the original depth map (Figure 5.3(g)), the depth maps recovered with the reference methods (Figure 5.3(h) to 5.3(j)), and the synthesised views using the proposed method (Figure 5.3(k)). These details clearly show the type of artefacts in synthesised views caused by an inaccurate recovered depth.

Thus, we can observe in Figure 5.3 the importance of recovering depth maps with high accuracy and the impact on the corresponding synthesised views. The detailed regions show that in presence of large depth variations, the proposed method is able to reconstruct the lost regions with very good quality. If thin objects exist in the image, like in the case of the legs of the ballerina, the proposed method has also good performance because it is able to efficiently match the boundaries of the candidate block with the undamaged neighbours. When observing the results obtained for *Packetization 2*, the proposed method achieves good results, even in large lost areas such as entire depth frames.

Figure 5.2 shows a PSNR plot for *Shark* (Figure 5.2(a)) and *Breakdancers* (Figure 5.2(b)) sequences, where the objective quality of each synthesised frame using the recovered depth maps can be seen. We can observe that using the proposed method the quality of the recovered frames is almost always higher than those obtained by the reference methods. The quality of synthesised frames can reach PSNR gains up to 2dB in the case of *Shark* sequence. For some frames of *Breakdancers* sequence the equivalent gain can be as high as 3dB.

Depth Maps:

(a) Original. (b) Corrupted. (c) Reference 1. (d) Reference 2. (e) Reference 3. (f) Proposed.

Synthesised images

:

(g) Original. (h) Reference 1. (i) Reference 2. (j) Reference 3. (k) Proposed.

**Figure 5.3** – Example of damaged and recovered depth maps with the corresponding synthesised images, sequence Ballet, PLR=10% using Packetization 2 (Frame 16).

## 5.2 Depth error concealment for multiple description decoders

As pointed out in Chapter 3, the impact of transmission errors and data loss is greater in 3D image and video communications than in traditional 2D communications, because the perceived quality of experience (QoE) is highly sensitive to a wider variety of quality factors [8]. In this context, multiple description coding (MDC) is an efficient approach to deal with error-prone multipath networks [183]. In MDC the source signal is encoded in two or more independent decodable streams (i.e., descriptions) which comprise correlated representations of the same source [184]. The great advantage of MDC is that a minimum signal quality is almost guaranteed because the probability of simultaneous loss of all descriptions is low in comparison with single stream coding and transmission, i.e., single description coding (SDC). However, the use of MDC reduces the coding efficiency. Yet, the improved robustness compensate for the loss of coding efficiency and the overall quality obtained after decoding MDC streams is better than SDC, especially for higher packet loss rates [185]. In this section the MDC approach is extended to depth maps, in the context of multiview video communications over multiple lossy channels.

In the past, some MDC methods have been proposed for 3D and multiview video communications. In [186], MDC is proposed for stereoscopic video based on spatial and temporal scaling. Temporal sub-sampling is also used in [187] based on a 3D even and odd MDC scheme with adaptive redundancy added to frames with motion activity higher than a threshold. An MDC scheme based on scalable coding was proposed in [188], where a redundant version of the enhanced layer is encoded as different description. In [189], an MDC scheme is presented using video-plus-depth coding, with viewpoint synthesis. Scalable coding is used and each of the two viewpoints is encoded with two spatial layers and two temporal layers. In addition to the scalable layers, a redundant stream for the base layer is generated by redundant encoding of the most significant foreground objects. A simulcast encoding scheme of temporal sub-sampled versions of each view are proposed in [190], using depth-image-based rendering (DIBR) to synthesize the missing view and motion compensation EC for occluded areas. In [191] a multiview MDC scheme based on spatial down sampling is proposed using multiview video coding amendment (MVC) of the H.264/AVC. All the previous works emphasize the effectiveness of MDC for 3D image and video signals in multipath video communications and its higher performance

in comparison with traditional SDC. However, delivery of depth maps encoded as multiple descriptions and transmitted through diverse error prone channels was not addressed.

The method described in this section deals with the problem of using MDC for multipath delivery of depth maps. Whenever any description is not available for decoding (e.g., due to packet loss), the resulting coarsely reconstructed values produce a significant amount of distortion in the synthesised images. Given the particular nature of depth information, i.e., smooth grey-level areas with reasonably well defined boundaries, a method to enhance the accuracy of depth maps decoded from one single description is proposed to improve the quality of synthesised images. Geometric information extracted from object boundaries of the depth map is used to compute new depth values, according to the relative spatial position of neighbour values.

### 5.2.1 Multiple description coding of depth maps

The MDC scheme used in this work to encode depth maps is based on Multiple Description Scalar Quantisation (MDSQ) to generate two independent descriptions from the same source signal, based on two functions: *scalar quantisation* and *index assignment* [192]. Scalar quantisation is generally applied to transform coefficients $x$, such that an index $i_1$ is produced for each one.

The index assignment process is based on an *index assignment matrix* (Table 5.3) to produce a side index pair $(i_1, i_2)$ from each central index $i_0$. Each side index corresponds to a quantisation cell containing several possible central indices. For instance, in Table 5.3, side index $i_1 = 2$ corresponds to a cell with central indices $i_2 \in \{4, 6, 7\}$. In general, MDC algorithms use balanced descriptions where the respective rates and distortions are approximately the same. Considering two descriptions with rate-distortion pairs $(R1, D1)$ and $(R2, D2)$, these two descriptions are generated from a central quantiser with rate-distortion pair $(R0, D0)$, using index assignment. This is equivalent to generating the two descriptions using coarser side quantisers. For balanced MDSQ, $R1 \approx R2$ and $D1 \approx D2$.

For a given source signal, an index assignment is defined to give the best row-column assignment of central quantisation indices along the matrix diagonal. For a selected set of side index pairs this should result in small spread in each cell. For a given index assignment matrix, the number of cells $(N)$ is determined by the number of different side

quantisation indices defined in that matrix. The central and side distortions $D0$ and $D1$ or $D2$, are defined as the distortion obtained from decoding both descriptions or only one of them, respectively. The side distortion is obviously higher than the central one and corresponds to the case where any single description is lost in the transmission path.

Both the side distortion and the amount of redundancy depend on the index spread ($k$), which represents the number of diagonals above (or below) the main diagonal in the index assignment matrix. Note that, central distortion $D0$ does not change with $k$ for a given fixed $N$, whereas side distortions $D1$, $D2$ increase with $k$. In the example of Table 5.3 $k = 1$, which results in 3 diagonals.

**Table 5.3** – Index assignment matrix ,$k$=1

| i1 (Description 1) \ i2 (Description 2) | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|---|---|---|
| -2 | | -8 | -6 | -7 | | | | | |
| -1 | | | -4 | -3 | -1 | | | | |
| 0 | | | | -2 | 0 | 1 | | | |
| 1 | | | | | 2 | 3 | 5 | | |
| 2 | | | | | | 4 | 6 | 7 | |

The MDC encoder includes the MDSQ module to produce two descriptions and the corresponding streams for transmission over different paths. The MDC encoding architecture uses both intra predicted and motion compensated predicted slices [185]. The headers, prediction modes and motion vectors are duplicated in both descriptions. In the decoder, if both descriptions are available, then an inverse index assignment process is used to restore a unique central index $i_0$ to be an inverse quantised and transformed. In the case where one description is not available for decoding due to transmission errors/losses, the other one is decoded using an equivalent coarser quantisation step size, which leads to a lower quality decoded slice. In Section 5.2.2, a method to enhance the quality of such slices is described.

### 5.2.2 Enhancement and error concealment of MDC slice based depth maps - Intra

The method proposed to enhance the quality of MDC depth maps decoded from only one MDSQ description comprises four processing steps as shown in Figure 5.4. Description losses are supposed to occur randomly at slice level, which means that not all slices of a

depth map are necessarily lost at the same time. If both descriptions are lost, then an EC method should be used in the decoder.

When only one description is lost, the first step of the enhancement process is to extract the contour information from depth slices affected by the description loss. Despite the low quality of slices decoded from single descriptions, in general, for the main objects in the 3D scene, it is still possible to extract depth contours with enough accuracy to be used by the enhancement process. The Canny Edge extraction algorithm is used to extract the depth map contours [112].

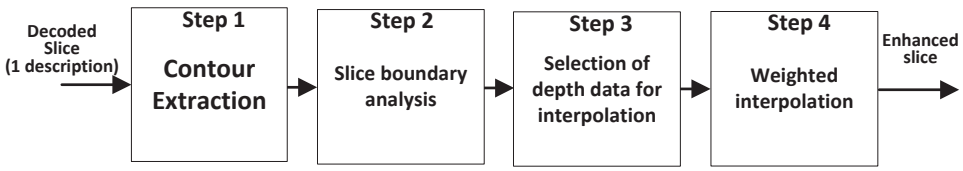



**Figure 5.4** – Enhancement diagram for MDC depth maps.

The second step consists in analysing the slice boundaries in order to find out whether a slice was decoded from one single description (low quality) and any of its neighbours from two descriptions (high quality). If there is a consistent difference between spatially adjacent slices, then an edge coincident with their common borders, should be found. In this case, the coarsely decoded depth values are enhanced through steps 3 and 4, as described in the following subsections.

### Depth data selection

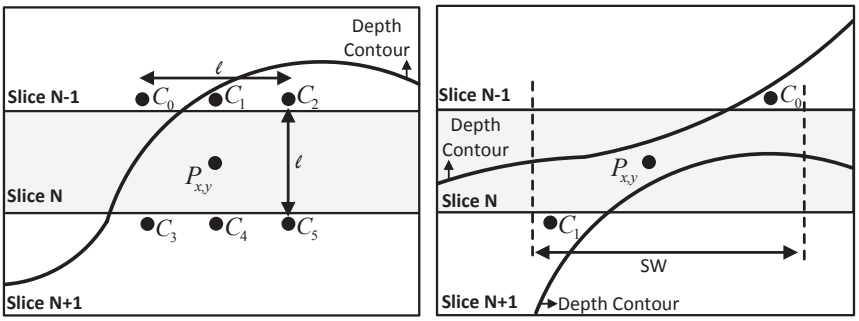
The object contours found in a single description slice are used to select the appropriate data to enhance each coarsely decoded depth value. Three different cases can be used, according to the available relevant data (Figure 5.5).

**Case I** In the first case, shown in Figure 5.5(a), if accurate neighbour depth values (i.e., a slice decoded from both descriptions) are available, then these are used to interpolate the coarse depth value ($P_{(x,y)}$). Up to six candidate depth values can be used: $C_0$ to $C_2$, from top slice N-1, and $C_3$ to $C_5$ from bottom slice N+1. If a certain candidate value is beyond the contour, then it is not used for interpolation because it belongs to a quite

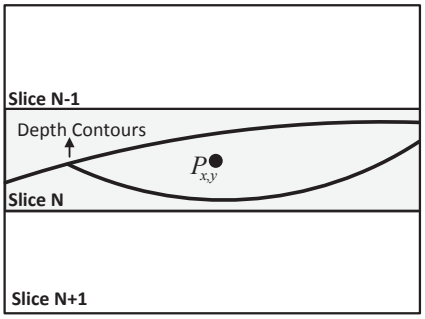
different depth level. If all six candidates are beyond the contour, then the coarse value fits into one of the next two cases.

**Case II** In the second case, shown in Figure 5.5(b), interpolation of the coarse value uses up to two neighbour candidates, one from the top slice ($C_0$) and another from the bottom slice ($C_1$). These values are found by searching in the region delimited by a predefined search window ($SW$). If this search is not successful in finding neighbour depth values belonging to the same region, i.e. delimited by the extracted contours, then the coarse depth value belongs to the third case.

**Case III** In the third case, shown in Figure 5.5(c), the coarse depth values usually belong to small object contours inside the coarsely decoded slice. Therefore, using neighbour data to enhance these depth values does not bring any guaranteed benefit. In this case the coarse depth value is not modified.




(a) Case I (b) Case II




(c) Case III

**Figure 5.5** – Example of using depth contours

### Weighted interpolation

After selecting the neighbour depth values ($C_i$), as described above, weighted interpolation is used to improve the accuracy of the coarse depth values. The following expression is used:

$$P'_{(x,y)} = \frac{P_{(x,y)} \times \frac{d_P}{0.5 \times l} + \sum_{i=1}^{N} C_i \times [1 - \frac{d_i}{D_{max}}]}{\frac{d_P}{0.5 \times l} + \sum_{i=1}^{N} (\frac{d_i}{D_{max}})}, \tag{5.3}$$

where $P_{(x,y)}$ and $P'_{(x,y)}$ are coarser and enhanced depth values, respectively, with coordinates $x$ and $y$. $l$ is the slice height, $d_p$ is the distance between $P(x, y)$ and the top or bottom limits of the slice being enhanced (the smallest of these two distances is chosen for $d_p$), $N$ is the number of values used for interpolation, $C_i$ is depth value $i$, $d_i$ is the distance between $P_{(x,y)}$ and $C_i$. Finally, $D_{max}$ is the maximum distance between depth values used in the interpolation.

## 5.2.3 Experimental results

The performance of the proposed method was evaluated by using two views of three MVD sequences with the same resolution, $1024 \times 768$: Ballet (view0,view2), Book Arrival (view8,view10) and Breakdancers (view0,view2).

Each sequence has different characteristics: Book Arrival has moderate object motion and complex depth structure, Breakdancers exhibits many objects in the scene (at different depths), and Ballet is less complex and has large objects. The reference software View Synthesis Reference Software (VSRS 3.5) was used to synthesise the intermediate view. The contours of depth maps were determined using the Canny algorithm implementation, as described in Section 4.2.2.

The texture (or colour) images and the corresponding depth maps used in the simulations were encoded at fixed QP=28 using H.264/AVC (reference software *JM17* [173]). One hundred frames were encoded using an IDR period of 12, and a GOP structure *IBPBP* with two reference frames. Slice mode was used and each depth map was divided into slices with a height of 64 pixels and width equal to the horizontal image resolution. Each

slice is packetized into one single packet. Two MDC balanced descriptions were used in the simulations, generated by an MDC encoder with 3 diagonals in index assignment matrices.

In these experiments, two independent network paths were simulated to deliver each description, each one with equal packet loss rate (PLR). A random packet loss generator with uniform distribution was used to drop packets according to the required PLR [180]. Transmission of each sequence of depth maps was simulated 10 times under the same PLR. Texture views are assumed to be transmitted without losses. As mentioned before, when both descriptions of a slice are simultaneously lost, a classic error concealment algorithm is used, either i) Spatial weighted interpolation, using the high quality depth values from error free neighbouring regions; ii) Motion-copy or iii) Region Copy, as defined in the reference software [173]. When only one description is lost, the proposed method is used to enhance the quality of the depth map decoded from one single description.

The algorithm was evaluated by computing the PSNR of synthesised views obtained from MDC depth maps transmitted through different channels, with random losses of 2%, 5%, and 10%. The reference view used for PNSR computation was synthesised using the uncompressed texture and depth maps. The quality of the synthesised views obtained from enhanced and not-enhanced depth maps is shown in Table 5.4, for the three concealment methods. The quality difference is shown in column $\Delta$PSNR. For reference, the PSNR of synthesised views obtained from error-free depth maps are also shown, i.e., PLR = 0%.

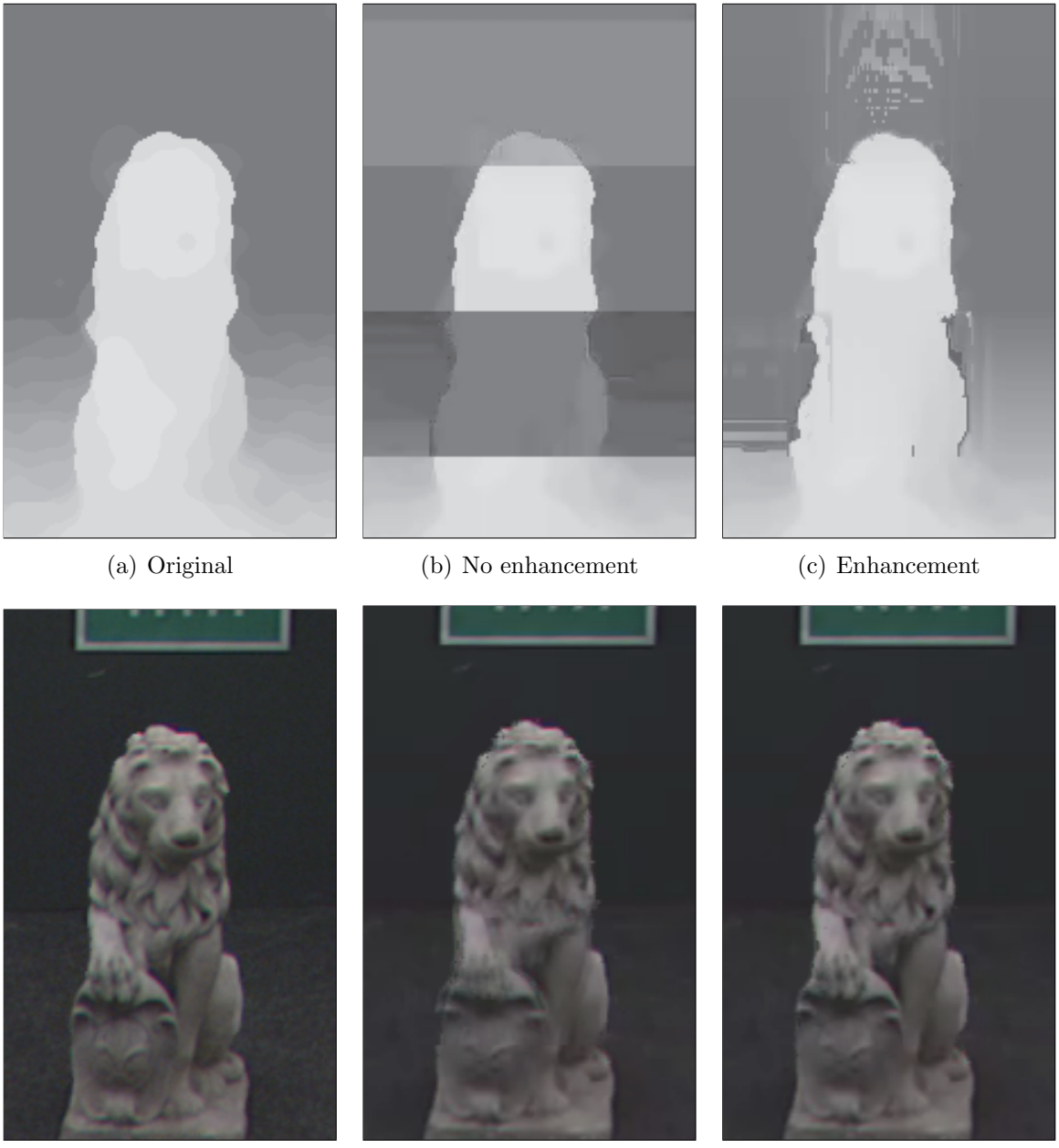
The simulation results show that the proposed method consistently enhances the reconstructed depth maps with positive impact on the quality of synthesised views. The higher gains occur for sequences with smooth areas and large objects, at different depth levels. Since lost descriptions correspond to an entire slice, which is a large area in the depth map, without the enhancement process, very often the synthesised images present poorer quality. Moreover, the distortion effect of inaccurate depth slices propagates throughout the GOP, contributing to additional degradation of the corresponding synthesised images. The maximum improvement in Table 5.4 is 1.69dB, obtained for *Ballet* sequence, for the worst case of 10% error loss. The proposed method is also able to achieve consistent quality improvement for different types of sequences and concealment methods.

These objective results can be subjectively confirmed, for instance, in Figure 5.6, where

**Table 5.4** – Average PSNR-Y of synthesised views

| | | Not Enhanced (a) | | | Enhanced (b) | | | ΔPSNR: (b)-(a) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| PLR | 0% | 2% | 5% | 10% | 2% | 5% | 10% | 2% | 5% | 10% |
| *Concealment method:* | **Spatial interpolation** | | | | | | | | | |
| Ballet | 35.99 | 34.69 | 33.61 | 32.25 | 35.24 | 34.77 | 33.90 | 0.55 | 1.16 | 1.65 |
| Book Arrival | 41.02 | 40.26 | 39.33 | 37.81 | 40.58 | 40.09 | 38.91 | 0.32 | 0.76 | 1.1 |
| Breakdancers | 36.34 | 34.94 | 34.06 | 32.53 | 35.18 | 34.39 | 33.32 | 0.24 | 0.33 | 0.79 |
| *Concealment method:* | **Motion copy** | | | | | | | | | |
| Ballet | 35.99 | 34.69 | 33.61 | 32.26 | 35.24 | 34.77 | 33.94 | 0.55 | 1.16 | 1.68 |
| Book Arrival | 41.02 | 40.26 | 39.33 | 38.87 | 40.58 | 40.11 | 38.99 | 0.32 | 0.78 | 1.1 |
| Breakdancers | 36.34 | 34.94 | 34.05 | 32.48 | 35.18 | 34.39 | 33.29 | 0.24 | 0.34 | 0.81 |
| *Concealment method:* | **Region Copy** | | | | | | | | | |
| Ballet | 35.99 | 34.69 | 33.61 | 32.25 | 35.24 | 34.77 | 33.94 | 0.55 | 1.16 | 1.69 |
| Book Arrival | 41.02 | 40.26 | 39.33 | 37.87 | 40.58 | 40.10 | 38.99 | 0.32 | 0.77 | 1.12 |
| Breakdancers | 36.34 | 34.94 | 34.05 | 32.49 | 35.18 | 34.39 | 33.90 | 0.24 | 0.34 | 0.81 |

the synthesised images are shown along with the respective depth maps. The example of Figure 5.6 shows a region detail of the original depth map (Figure 5.6(a)), the coarsely decoded depth map with 10% of loss (Figure 5.6(b)), and the enhanced depth map using the proposed method (Figure 5.6(c)). The corresponding synthesised views using the original depth map are shown in Figure 5.6(d), the synthesised view using the non-enhanced depth map is shown in Figure 5.6(e) and the synthesised view using the proposed enhancement method is shown in Figure 5.6(f). These detailed regions show that when there are large depth variations, the proposed method is able to reconstruct the lost regions with very high accuracy. In the presence of thin objects fully included in the coarse area, it is much more difficult to improve the depth map quality.



(a) Original (b) No enhancement (c) Enhancement

(d) Original (e) No enhancement (f) Enhancement

**Figure 5.6** – MDC depth maps and respective synthesised views at 10% of packet loss (frame 84).

## 5.3 Enhancement and error concealment of MDC slice based depth maps - Intra and Inter techniques

In this section depth maps are encoded with MDC using the coding scheme described in Section 5.2. The improvement over the previous technique is focused on enhancing the spatial interpolation technique and exploiting the temporal information of the corresponding texture image of the corrupted depth map. The usage of temporal information allowed this EC technique for MDC depth maps to be used even when the description of the entire frame is lost. Thus, this new technique becomes more versatile, allowing to better recover lost, descriptions representing only small regions of a depth maps (eg. Slices) up to the entire frame.

When both descriptions of a slice are simultaneously lost *Full reconstruction* is used, based on a classic error concealment method, either i) *Spatial Interpolation* - using high quality depth values from error free neighbouring regions; ii) *Motion Vector Sharing* or iii) *Frame Copy* - as defined in the reference software [173]. However, *Spatial Interpolation* cannot be used if the whole depth map is lost. But, when only one description is lost the proposed reconstruction method uses information from the received description and from the texture frame, as described in the folowing.

The proposed MDC depth reconstruction algorithm is shown in Figure 5.7. When a depth region is just coarsely decoded due to the loss of one description, the first step is to extract geometric information by computing the existing edges of the received depth description, using the *Canny* edge detection algorithm [112]. The next step is to identify the existence of motion information corresponding to the lost region. If the lost depth region is intra-coded, as well as, the corresponding texture, then motion information does not exist, therefore only spatial error interpolation is used (Section 5.3.2). On the other hand, if the corresponding texture is inter-predicted, its motion information can be used for temporal reconstruction of lost depth. However, temporal reconstruction is only performed if accurate motion vectors exist, which implies that usually some depth values are not temporally reconstructed. In such case, as explained in Section 5.3.1, spatial interpolation is used. Accurate motion vectors are defined as those associated with blocks with null residue, meaning a perfect match with the temporal reference.

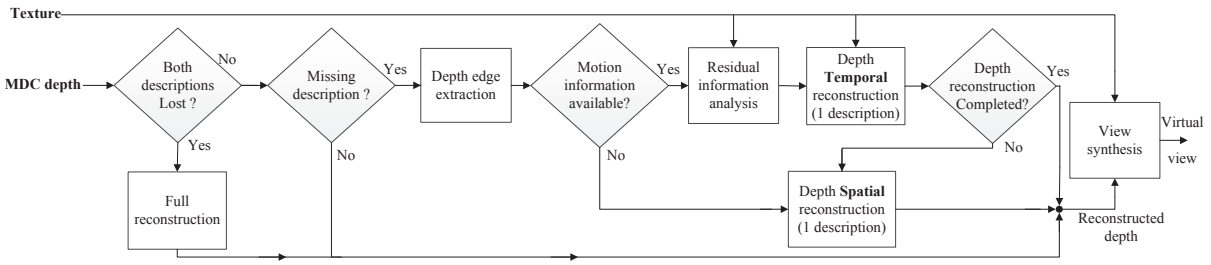



**Figure 5.7** – MDC depth map reconstruction algorithm and view synthesis.

### 5.3.1 Temporal reconstruction

As shown in the block diagram of Figure 5.7, the existence of accurate motion information allows to reconstruct the corresponding lost depth. If the depth slice of a lost description is inter-coded, then temporal reconstruction using motion vectors from the other description is, in general, very accurate because the residue is zero for many slice blocks. The most challenging case occurs when the lost description is intra-coded or when it also includes intra-coded blocks. Since there are not motion vectors in the other description, the texture is used to provide the necessary motion information. As previously described, only motion vectors associated to zero residue are used because they correspond to the most accurate motion. This condition prevents geometric distortions in the reconstructed depth that would appear otherwise, with serious negative impact on synthesised view quality. Thus, after motion compensation of previously decoded depth maps using accurate motion vectors, the remaining missing depth values are interpolated, as described in Section 5.3.2. Note that, temporal domain reconstruction is performed prior to spatial domain interpolation, in order to provide accurate depth values for spatial interpolation afterwards.

The GOP structures used for texture and depth are unaligned and their sizes are co-prime numbers to maximise the temporal distance between frame instants, where simultaneous intra-coding occurs in both texture and depth (e.g., in these frame instants no motion information exists from texture neither from depth descriptions). In this work an IBPB structure was used and the GOP sizes were defined as 11 and 13 for texture and depth, respectively, as shown by Figure 5.8.

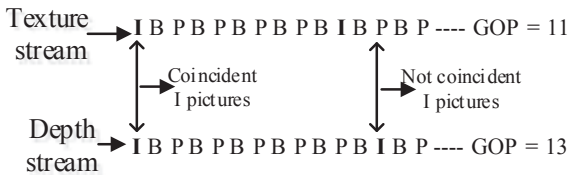



**Figure 5.8** – Texture and depth GOP structure.

### 5.3.2 Spatial interpolation

Spatial interpolation of missing depth is performed in two cases: i) when no motion information is available and ii) where the missing depth values are not temporally reconstructed. The spatial interpolation process uses neighbouring depth values, which are carefully chosen around the value being reconstructed, taking into account the geometric features of the surrounding depth region, for instance edges.

Figure 5.9 shows how the neighbour values $P_i$ are determined for interpolation of the coarsely decoded depth value $Y_{(x,y)}$. A set of points is chosen around the spatial position $(x, y)$, using $N$ directions within a predetermined search window $SW$. For each direction, the nearest depth value that meets two criteria is chosen: 1) $P_i$ is located in the same region of $Y_{(x,y)}$, which is delimited by the extracted depth contours; 2) $P_i$ belongs to the set of accurate depth values, which comprises those correctly received from both descriptions and also the ones temporally reconstructed. In this work, 16 directions were uniformly defined around (x,y), i.e., N=16.

After selecting all possible $P_i$, reconstruction of the coarsely decoded value $Y_{(x,y)}$ is done by computing a new depth value $Y'_{(x,y)}$ using weighted interpolation, as defined by Equation 5.4,

$$Y'_{(x,y)} = \frac{Y_{(x,y)} \times \frac{d_{min}}{0.5 \times SW} + \sum_{i=0}^{N} P_i \times [1 - \frac{d_i}{2 \times d_{max}}]}{\frac{d_{min}}{0.5 \times SW} + \sum_{i=0}^{N} [1 - \frac{d_i}{2 \times d_{max}}]}, \tag{5.4}$$

where $(x, y)$ is the coordinate of the depth map value being interpolated, $d_i$ is the distance between $(x, y)$ and $Pi$, $d_{min}$ and $d_{max}$ are the minimum and maximum distances in $[d_i]$, with $i = 1...N$. The weighting factor of $Y_{(x,y)}$ depends on $d_{min}$ and the size of the

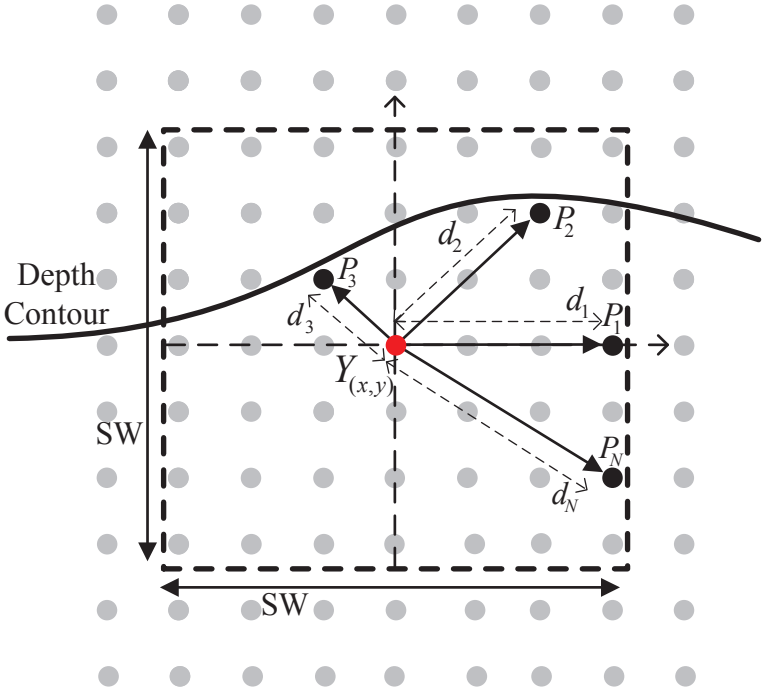



**Figure 5.9** – Selection of depth values for interpolation.

search window $SW$. It gives less or more importance to the coarsely decoded value $Y_{(x,y)}$, depending on the $P_i$ values in the neighbourhood of $(x, y)$. The weight of $Pi$ depends on its distance $di$ and $d_{max}$, giving more relevance to the $Pi$ values close to $(x, y)$.

The performance of the proposed method was evaluated through the PSNR of virtual views, synthesised from MDC depth streams transmitted over different simulated paths, subject to random losses of 2%, 5%, and 10%. The reference virtual view used for computing the PSNR was synthesised from uncompressed texture and depth.

### 5.3.3 Experimental results

The following sequences and views were used: Ballet (views 0,2), Book Arrival (views 8,10) and Breakdancers (views 0,2), with spatial resolution of $1024 \times 768$ pixels and one hundred frames, Shark (views 1,5) with $1920 \times 1088$ pixels and three hundred frames. These four sequences were chosen due to their different characteristics, namely spatial resolution and different types of objects in the scene, with various sizes and shapes. These characteristics correspond to depth maps with different features, which is useful to validate the results.

Note that, the first three sequences are natural scenes obtained from a camera and Shark is a computer-generated sequence.

The reference software VSRS [29] was used to synthesise the virtual views. Edges were extracted from the depth maps using the OpenCV [177] implementation of the Canny edge extraction algorithm. Texture and depth were encoded using H.264/AVC, fixed QP=28 and GOP structure *IBPBP..* using reference software JM17 [173]. As in Section 5.3.1, texture and depth sequences were encoded using different GOP sizes, respectively, 11 and 13. Two balanced descriptions were generated by an MDC encoder using index assignment matrices with 3 diagonals. Packet losses were simulated for each depth stream description of the second view of each sequence.

Two different encoding/packetisation modes were used: *Slice mode* and *Full frame*. In the former, each slice is packetized into one single packet, being each depth map is divided into slices of eight=64 pixel and width equal to the horizontal picture resolution. In the latter, *Full frame*, each slice corresponds to an entire depth frame, which is then packetized into individual packets.

In these experiments, two independent network paths are simulated to deliver each description subject to the same PLR. A random packet loss generator with uniform loss probability distribution was used to drop packets according to predefined values of PLR. Transmission of each depth stream was simulated 30 times under the same PLR for both modes: *Slice mode* and *Full frame*.

The average PSNR obtained for the various virtual views is shown in Table 5.5. In this table, column *Not Enhanced (b)*, represents the case where reconstruction is performed only when both descriptions are lost. Column *Proposed (c)* presents the PSNR obtained for the proposed reconstruction method using information from a single description. The results are shown for different *Full reconstruction* methods, which are complementary to the proposed one, since they are only used when both descriptions are simultaneously lost.

It is necessary to evaluate whether different *Full reconstruction* methods have any impact on these results. The quality difference between the virtual views synthesised with the proposed method and the reference one (*Not Enhanced (b)*) is shown in column $\Delta$PSNR. For reference, the average PSNR difference between each lossy case (PLR 2% to 10%)

**Table 5.5** – Average PSNR-Y of the synthesised views.

| | | (a) | Not Enhanced (b) | | | Proposed (c) | | | ΔPSNR: (c)-(b) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | PLR: | 0% | 2% | 5% | 10% | 2% | 5% | 10% | 2% | 5% | 10% |
| *Slice mode* | *Full reconstruction* method: | **Spatial Interpolation** | | | | | | | | | |
| | Ballet | 36.14 | 34.44 | 33.89 | 31.96 | 35.28 | 34.98 | 33.91 | 0.84 | 1.09 | 1.95 |
| | Book Arrival | 41.15 | 40.34 | 39.30 | 37.87 | 40.73 | 40.21 | 39.39 | 0.39 | 0.91 | 1.52 |
| | Breakdancers | 36.37 | 34.75 | 33.84 | 32.68 | 34.94 | 34.60 | 33.93 | 0.19 | 0.76 | 1.25 |
| | Shark | 40.78 | 38.87 | 37.97 | 36.72 | 39.28 | 38.79 | 37.94 | 0.50 | 0.82 | 1.22 |
| | Average PSNR diff: (a)-(b) or (a)-(c) | | 1.51 | 2.36 | 3.80 | 1.01 | 1.46 | 2.32 | – | – | – |
| | *Full reconstruction* method: | **Motion Vector Sharing** | | | | | | | | | |
| | Ballet | 36.14 | 34.46 | 33.90 | 32.08 | 35.30 | 35.02 | 34.37 | 0.84 | 1.12 | 2.29 |
| | Book Arrival | 41.15 | 40.34 | 39.31 | 37.91 | 40.74 | 40.31 | 39.55 | 0.40 | 1.00 | 1.64 |
| | Breakdancers | 36.37 | 34.74 | 33.84 | 32.69 | 34.94 | 34.60 | 33.95 | 0.20 | 0.76 | 1.26 |
| | Shark | 40.78 | 38.78 | 37.97 | 36.73 | 39.28 | 38.78 | 37.94 | 0.50 | 0.81 | 1.21 |
| | Average PSNR diff: (a)-(b) or (a)-(c) | | 1.53 | 2.35 | 3.76 | 1.04 | 1.43 | 2.16 | – | – | – |
| | *Full reconstruction* method: | **Frame Copy** | | | | | | | | | |
| | Ballet | 36.14 | 34.46 | 33.88 | 32.07 | 35.30 | 35.02 | 34.35 | 0.84 | 1.14 | 2.28 |
| | Book Arrival | 41.15 | 40.34 | 39.30 | 37.89 | 40.73 | 40.30 | 39.53 | 0.39 | 1.01 | 1.63 |
| | Breadancers | 36.37 | 34.73 | 33.83 | 32.68 | 34.94 | 34.60 | 33.93 | 0.21 | 0.77 | 1.25 |
| | Shark | 40.78 | 38.77 | 37.96 | 36.71 | 39.28 | 38.78 | 37.92 | 0.51 | 0.82 | 1.21 |
| | Average PSNR diff: (a)-(b) or (a)-(c) | | 1.53 | 2.37 | 3.77 | 1.05 | 1.43 | 2.18 | – | – | – |
| *Full frame* | *Full reconstruction* method: | **Motion Vector Sharing** | | | | | | | | | |
| | Ballet | 36.14 | 35.12 | 34.28 | 33.80 | 35.43 | 35.16 | 34.76 | 0.31 | 0.88 | 0.96 |
| | Book Arrival | 41.15 | 40.54 | 40.27 | 39.32 | 40.69 | 40.55 | 40.19 | 0.15 | 0.28 | 0.87 |
| | Breakdancers | 36.37 | 35.24 | 34.62 | 33.74 | 35.37 | 34.88 | 34.32 | 0.13 | 0.26 | 0.58 |
| | Shark | 40.78 | 39.25 | 39.06 | 37.66 | 39.39 | 39.28 | 38.33 | 0.14 | 0.22 | 0.67 |
| | Average PSNR diff: (a)-(b) or (a)-(c) | | 1.07 | 1.55 | 2.48 | 0.89 | 1.14 | 1.71 | – | – | – |
| | *Full reconstruction* method: | **Frame copy** | | | | | | | | | |
| | Ballet | 36.14 | 35.12 | 34.28 | 33.80 | 35.42 | 35.16 | 34.76 | 0.29 | 0.88 | 0.96 |
| | Book Arrival | 41.15 | 40.54 | 40.26 | 39.32 | 40.69 | 40.53 | 40.18 | 0.15 | 0.27 | 0.87 |
| | Breakdancers | 36.37 | 35.25 | 34.62 | 33.73 | 35.37 | 34.88 | 34.31 | 0.12 | 0.27 | 0.58 |
| | Shark | 40.78 | 39.25 | 39.06 | 37.65 | 39.39 | 39.30 | 38.33 | 0.14 | 0.24 | 0.65 |
| | Average PSNR diff: (a)-(b) or (a)-(c) | | 1.07 | 1.55 | 2.48 | 0.89 | 1.14 | 1.71 | – | – | – |

and lossless one (0%) is shown in Table 5.5.

The PSNR results presented in Table 5.5 show that the proposed method consistently outperforms the classic concealment method across a wide range of PLR, even when large depth areas are corrupted, which is the case of mode *Full frame*. The good performance in large corrupted areas is due to the combined use of temporal reconstruction followed by spatial interpolation. In larger error areas, the use of depth values obtained from previously decoded depth maps (i.e., temporal reconstruction) is essential for providing accurate depth values for the subsequent spatial interpolation. This allows the proposed method to consistently achieve higher PSNR than case *b)*, with a difference up to 2.29dB, at 10% PLR in sequence *Ballet* (*Slice mode*). The average PSNR difference confirms that the proposed method always results in better virtual views for all sequences, regardless of the full reconstruction method in use, i.e., (a)-(c) is always lower than (a)-(b).

Simple observation of the results presented in Figure 5.10 confirms the importance of accurate depth map reconstruction and the better performance of the proposed method. The loss of one description seriously affects the depth map quality, as can be seen in Figure 5.10(b). The proposed method significantly enhances the coarsely decoded depth map (Figure 5.10(c)) resulting in a synthesised image with significantly higher quality (Figure 5.10(f)). In fact, there are many annoying artefacts in Figure 5.10(e), obtained with a reconstructed depth map which was decoded without enhancing after losing one description (Figure 5.10(b)). The virtual view obtained from the proposed method (Figure 5.10(f)) is much closer to the one synthesised with an error-free depth map (Figure 5.10(d)). Similar consistent results are obtained for all sequences.

To conclude, this is an efficient reconstruction method for MDC depth maps, which can be used when any description is lost. The experimental results show that the proposed method is very efficient, not only when applied to small depth error regions, but also in large error areas, even when an entire depth frame is lost. Experimental results obtained with the proposed method show that the use of motion information from the texture clearly improves the reconstruction performance. Furthermore, the combination of temporal and spatial techniques results in more accurate MDC depth map values, significantly improving the quality of the synthesised views.

(a) Original. (b) Not Enhanced. (c) Proposed.

(d) Original. (e) Not Enhanced. (f) Proposed.

**Figure 5.10** – MDC depth maps and respective synthesised views at 10% of packet loss (frame 12), Slice mode.

# 6 Perceptually-aware quality metric for performance evaluation of depth error concealment

This chapter presents a study about quality evaluation of the error concealment methods for depth maps used in multiview video-plus-depth (MVD). The main research objective was to evaluate the performance of depth error concealment methods using an objective approach, but including perceptual factors in the quality assessment. The performance of two error concealment methods for depth maps is evaluated using a perceptually-aware objective metric. This metric is validated through subjective assessment of virtual views synthesised with recovered depth maps. The perceptual impact of reconstruction in corrupted depth of MVD is evaluated under various loss rates, using several texture images and depth maps encoded at multiple quantisation steps. The results revealed that the proposed objective quality metric is mostly inline with user preferences, in respect to the relative performance of each error concealment method.

## 6.1 Quality evaluation of synthesised views

As mentioned before, the perceived video quality in MVD relies on the synthesised views, which is related to the quality of the decoded depth maps. In general, video quality metrics are more sensitive to capture coding distortions, which are not of the same type as those resulting from reconstruction of depth maps using error concealment methods. This particular type of distortions mostly affect geometric characteristics of the synthesised views and objects in the scene. Therefore, the traditional objective quality metrics,

normally used to evaluate the quality of texture images, may not capture all the perceptually relevant artefacts created by error concealment methods. This leads to a challenging problem associated with view synthesis in the MVD format, which is the choice of the best quality evaluation methods for synthesised images. Recent studies indicate that current 2D video objective quality metrics might not be suitable to directly evaluate the quality of synthesised images [193]. Thus, reliable metrics for evaluation of synthesised views, specifically tailored for depth maps, is still an open problem. Nevertheless, in the literature there are some research works covering the topic of quality evaluation of synthesised views and relating objective and subjective assessment. In [194, 195] specific types of artefacts occurring in DIBR are studied and the authors provide a contribution to develop objective metrics, resulting in an objective metric correlated with the existing subjective assessment methods.

Objective metrics usually tend to show low correlation with subjective measures [196] in the case of distortion caused by encoding and by DIBR algorithms. Correspondence of subjective scores with objective measures, such as PSNR and SSIM, was not yet addressed in the literature considering depth maps recovered from transmission losses. The objective of this work is to address this specific case of quality evaluation by studying the effect of two spatial error concealment methods. The first method interpolates lost depth values using the neighbouring ones and the second method was described in Section 4.5. Using these two EC methods for different error rates and different data loss patterns, a subjective and objective quality evaluation study was performed, in order to find a relation between the different approaches of synthesised images when using recovered depth maps.

### 6.1.1 Methods used for performance evaluation

As mentioned above, two error concealment methods were used to recover corrupted depth maps. The first method (ECM1) is based on spatial error concealment using weighted interpolation to fill the missing regions, while the second one (ECM2) was proposed in Section 4.5. Neither of theses methods use temporally adjacent depth maps for concealment.

ECM1 is commonly used in H.264/AVC decoders. It is based on the weighted interpolation of the missing regions, using the neighbour values correctly received and decoded. Depending on their availability, up to four neighbour values can be used to recover one

single missing depth value of the lost region, e.g., from the top, bottom, left and right.

ECM2 exploits depth discontinuities, besides the neighbouring depth values as ECM1 and inter-view information through geometric transforms. Temporal information is not used, since this error concealment method was designed for intra frame coding, where temporal information is not available (see Section 4.5).

In this work, a pair of intra-coded views from different sequences and the corresponding depth maps are used to evaluate the performance of the two error concealment methods, measured by the quality of the corresponding synthesised views. The slice arrangement and packetization define the structure of the lost regions, which consist in either square areas (using Flexible Macroblock Ordering - FMO) or entire rows of macroblocks, dispersed throughout the depth map. More details about these error patterns and PLRs are given in Section 6.3.

## 6.2 Perceptually-aware objective metric

The proposed objective quality metric is based on the PSNR and it is a full reference method, where the virtual views, synthesised with the original texture and depth maps are used as reference. The proposed metric takes into account the regions with different perceptual relevance in the synthesised images. Such regions are determined based on the reconstruction accuracy of the concealment method and the higher the perceptual relevance of depth discontinuities (i.e., edges). The underlying idea is that inaccurate concealment of such depth regions leads to annoying distortions, which are more likely to cause visual discomfort. The metric proposed in this work is computed in four steps, measuring the distortion of a generic view $N$ synthesised from views $N-1$ and $N+1$ (texture plus depth).

• **Step 1:** The first step computes a binary mask $B_N(x, y)$, representing regions where view synthesis were not accurately computed, due to non-perfect concealment of the corrupted depth map regions. This is done by computing the absolute differences between views synthesised with the recovered depth maps and error-free depth map (i.e., the reference). The binary mask $B_N(x, y)$ is given by Eq. 6.1 for each spatial position ($x$,$y$).

$$B_N(x,y) = \begin{cases} 1, & \text{if } |V_N^e(x,y) - V_N^c(x,y)| \neq 0 \\ 0, & otherwise \end{cases} \tag{6.1}$$

$V_N^e(x,y)$ and $V_N^c(x,y)$ are the synthesised views $N$ from the error-free and recovered depth maps, respectively.

• **Step 2:** In the second step, a binary map $E_c$ is computed, considering only the most perceptually important pixel positions $(x,y)$, i.e., those that are more likely to cause annoyance to viewers. To this end, an edge mask $E(x,y)$ is first computed for the error-free depth map using the *Canny* algorithm. Then, the edge mask of either depth map $N-1$ or $N+1$ is matched with the view that is being synthesised, i.e., view $N$. This is done by computing the disparity map $D(x,y)$ between views $N-1$ and $N+1$ using the algorithm proposed in [178]. The new disparity map $D_N(x,y)$ for view $N$ is computed using Eq. 6.2.

$$D_N(x,y) = D(x,y) - \frac{D(x,y) * d(N,N-1)}{d(N-1,N+1)} \tag{6.2}$$

where $d(N,M)$ is the distance between two cameras, N and M. The disparity compensated edge mask $E_c(x,y)$ is computed for view $N$, as shown by Eq. 6.3.

$$E_c(x,y) = E(x + D_N(x,y), y) \tag{6.3}$$

Finally, a morphological dilation is performed on $E_c(x,y)$ using a square window with size $W$, in order to expand the surrounding areas of the edges. As mentioned before, these regions are considered to have a higher perceptual relevance.

• **Step 3:** The objective of the third step is to generate two different binary masks, $B_N^1(x,y)$ and $B_N^2(x,y)$ identifying, respectively, the most and the least perceptually important regions of the synthesised view $N$. Note that this perceptual importance only distinguishes regions where non-accurate EC was performed.

$B_N^1(x,y)$ is obtained from a logical operation ($B_N(x,y)$ AND $E_c(x,y)$), as well as $B_N^2(x,y)$, as shown by Equation 6.5.

$$B_N^1(x,y) = B_N(x,y) \;\&\; E_c(x,y) \tag{6.4}$$
$$B_N^2(x,y) = B_N(x,y) \;\&\; \overline{B_N^1(x,y)} \tag{6.5}$$

$B_N^2(x,y)$ defines the pixel locations where $V_N^e(x,y)$ differs from $V_N^c(x,y)$ and a depth discontinuity does not exists.

• **Step 4:** The final step performs the computation of the perceptually-driven PSNR (pPSNR) and MSE (pMSE), as defined by Equation 6.6, using different weights, $w_1$ and $w_2$, to diferentiate the importance of each region.

$$\begin{aligned} pPSNR &= 10\log_{10}\left(\frac{255^2}{pMSE}\right) \\ pMSE &= \frac{MSE_1.w_1 + MSE_2.w_2}{w_1 + w_2} \\ MSE_i &= \frac{1}{\Sigma B_N^i}\sum_{x,y}(V_N^e(x,y) - V_N^c(x,y))^2 B_N^i(x,y), i = 1,2 \end{aligned} \tag{6.6}$$

## 6.3 Experimental setup

The performance of the error concealment methods ECM1 and ECM2 is evaluated through objective and subjective tests. It assesses how a single corrupted depth map affects the virtual views, synthesised from its recovered version, using ECM1 and ECM2. Since both ECM1 and ECM2 do not use temporal information in the concealment, only spatial and inter-view information is used. Thus, only one temporal instant (two images and their depth maps) of each sequence is used, as shown in Table 6.1.

**Table 6.1** – Used images and the corresponding characteristics.

| Sequence | Resolution | Left View | Right View | Frame num. |
|---|---|---|---|---|
| 1-Book Arrival | 1024×768 | camera 6 | camera 10 | 1 |
| 2-Newspaper | 1024×768 | camera 2 | camera 6 | 1 |
| 3-Shark | 1920×1088 | camera 1 | camera 5 | 249 |
| 4-Dancer | 1920×1080 | camera 1 | camera 9 | 249 |

For each sequence, one view is selected from two different cameras (ground truth). Using these two views, 49 equidistant intermediate viewpoints are synthesised from left to right, and another 49 equidistant viewpoints from right to left in order to create a video sequence of 100 frames. The first frame (ground truth) is the first view, followed by 49 synthesised ones. The 51st frame is the second view acquired by the second camera (ground truth), followed by the same 49 synthesised views in the other direction (from the second view to first the view). The reference software VSRS-1D-FAST was used to synthesise the intermediate views.

In order to evaluate and compare methods ECM1 and ECM 2, the transmission process of texture and depth maps of both views is simulated using the following steps: coding, transmission, error simulation/decoding and finally, the synthesis of intermediate views. Transmission errors are simulated in the depth map of the second view, while the texture of both views and the depth of the first view are received and decoded without errors/losses. The texture images and depth maps were encoded with H.264/AVC, using the reference software JM17, and three quantisation steps (QP): 28, 34 and 40. To simulate the lost regions in the depth map of the second view, two error patterns were used. The first one (*Error Pattern 1*) corresponds to square blocks ($64 \times 64$), equally spaced in the depth map, and the second error pattern (*Error Pattern 2*) is comprised of stripes with a height of 64 depth values and width equal to the horizontal resolution of the image. As mentioned before, these error patterns are likely to occur when using H.264/AVC slice mode or Flexible Macroblock Order (FMO). A lost packet might correspond to the loss of one or more slices or slice groups (FMO). Figure 6.1 shows an example of a corrupted depth map with the packet loss rate (PLR) of 20% for *Error Pattern 1* (Figure 6.1(a)) and *Error Pattern 2* (Figure 6.1(b)).

### 6.3.1 Subjective Evaluation

Table 6.2 shows the parameters used in the subjective tests. Each row corresponds to a comparison pair for which a subjective score is obtained. For each QP, ten different comparisons were performed. In the column *Test name* "sq" refers to the squares error pattern (*Error Pattern 1*) and "st" refers to the stripes error pattern (*Error Pattern 2*). The first comparison corresponds to the synthesised video using the error free decoded depth map versus the one recovered with the ECM 2 with *Error Pattern 2* and ER =

(a) Error Pattern 1. (b) Error Pattern 2.

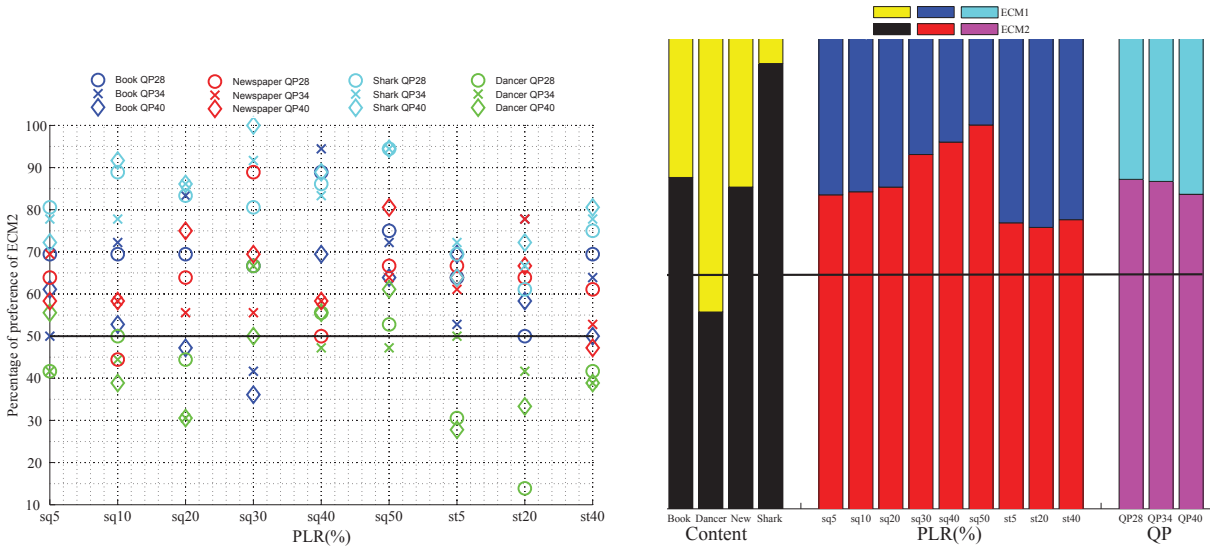
**Figure 6.1** – Corrupted depth map, Book Arrival, PLR of 20%.

**Table 6.2** – Test cases for the subjective assessment

| Test name: | QP=28;34;40; | | | | |
|---|---|---|---|---|---|
| sq0 | Error free | | vs | Error P. 1; 5%; | ECM1 |
| sq5 | Error P. 1; 5%; | ECM1 | vs | Error P. 1; 5%; | ECM2 |
| sq10 | Error P. 1; 10%; | ECM1 | vs | Error P. 1; 10%; | ECM2 |
| sq20 | Error P. 1; 20%; | ECM1 | vs | Error P. 1; 20%; | ECM2 |
| sq30 | Error P. 1; 30%; | ECM1 | vs | Error P. 1; 30%; | ECM2 |
| sq40 | Error P. 1; 40%; | ECM1 | vs | Error P. 1; 40%; | ECM2 |
| sq50 | Error P. 1; 50%; | ECM1 | vs | Error P. 1; 50%; | ECM2 |
| st5 | Error P. 2; 5%; | ECM1 | vs | Error P. 2; 5%; | ECM2 |
| st20 | Error P. 2; 20%; | ECM1 | vs | Error P. 2; 20%; | ECM2 |
| st40 | Error P. 2; 40%; | ECM1 | vs | Error P. 2; 40%; | ECM2 |

5%. For all other comparisons, the test is performed between both methods. Six PLRs were used for Error Pattern 1 (5%, 10%, 20%, 30%, 40% and 50%). In order to reduce the number of comparisons in the subjective evaluation, only three PLRs were used for Error Pattern 2 (5%, 20% and 40%). Ten tests for each QP and for each content (See table 6.1) were made, resulting in 120 comparisons.

The subjective experiment was conducted in an ITU-R BT.500-11 conforming subjective test environment. The stimuli were displayed on 2 Philips 46PFL9705H displays in a time parallel pair comparison. This is different from the standard ITU-T P.910 where images are presented one after each other. Only one observer at a time participated in the experiment. The observers had the possibility to replay the videos as much as needed. The presentation order (conditions and ECM) was randomized for each observer. The randomization process allows to globally balancing the number of visualizations on each display (to avoid a display preference). Thirty-six naive observers evaluated all 120 videos within a time frame of about 30 minutes.




(a) Preference percentage of ECM2.

(b) Observers mean preference.

**Figure 6.2** – Subjective results obtained from the 36 observers.

Figure 6.2 presents two charts regarding the subjective results obtained from the group of observers opinion. Figure 6.2(a), show the observer preference percentage between ECM2 and ECM1. While for *Dancer* view-sequence the method preference is around 50% (sq5 to sq50), meaning that both ECM performs in a similar manner, for other view-sequences, it is clear that ECM2 is preferred. For *Dancer*, in cases st5, st20 and st40(the Error P. 2 pattern), ECM1 is preferred. It is also shown in Figure 6.2(b) that for *Dancer* ECM1 is preferred (44% for ECM2), but for the remaining content, ECM2 is always preferred (64%, 63% and 81%).

In Figure 6.2(b), we can observe that the ECM2 preference percentage depends on the PLR for *Error Pattern 1* (from 62% for ER=5% to 70% for ER=50%). For *Error Pattern 2*, there is almost no percentage variation (between 57% and 58% in any case). Finally, we can see the percentage repartition does not depend on the QP values. From 64% for QP=28 to 62% for QP=40, which leads to the conclusion that the quantisation step affects the concealment process of ECM1 and ECM2 equally, because the choice of the observers remains similar over the tested QP's.

A statistical study was also performed with *Chi-Square* analysis [197], by using each of the available "elements" (content, QP values and PLRs). This analysis was performed in order to understand if the observers preference percentage between the two error concealment

methods depend on the contents, QP values and PLR. The *Chi-Square* analysis shows in charts of Figure 6.2 that the preference percentage does not depend of the QP. The preference percentage depends on the contents used and the PLRs for *Error Pattern 1* (sq5 to sq50). For Error P.2 (st5, st20 and st40), the preference does not depend on the PLR, which means that for larger areas using *Error Pattern 1* the concealment performance of the two methods is affected in a similar manner by each PLR.

The binomial distribution of the same available elements (content, QP values and PLRs) was also performed. It was found, that with the number of tests performed for each comparison, the probability of having equal results in the two error concealment methods is null. This means that all preference percentage bars show a relevant difference between the two error concealment methods. ECM2 method is always the preferred one, except in the case of *Dancer* where ECM1 performs slightly better.

### 6.3.2 Objective Evaluation *vs.* Subjective evaluation.

For objective evaluation three metrics were used: Peak Signal to Noise Ratio (PSNR), Structure Similarity (SSIM) and the proposed pPSNR. The reference used for comparison was the sequence of one-hundred intermediate views, as described in Section 6.3.1, using both original texture images and the corresponding depth maps (not encoded).

To use the proposed quality metric (pPSNR), the following parameters were used: The window size W=4 (see Step 2 in Section 6.2) and the weights $w_1 = 1$ and $w_2 = 0.5$, giving more weight to the regions around depth discontinuities (defined by $B_N^1(x, y)$), in comparison to other regions affected by distortions in the depth maps, defined by $B_N^2(x, y)$.

Table 6.4 shows the PSNR and SSIM results for the same synthesised videos, as used in the subjective evaluation. The PSNR for the ECM1 is generally higher than ECM2, especially in the cases where the depth map is subject to higher loss rates and when the depth map presents more shapes at different depths, resulting in a depth map less homogeneous. In these cases ECM2 shows higher PSNR results because it was designed to preserve more efficiently the depth shapes resulting in improved performance, when compared to ECM1. For instance, the PSNR of ECM2 can be higher than ECM1 up to 1.59dB in *Book Arrival* (QP=28, ER=20% using Error P. 1).

ECM1 only performs better than ECM2 in the case of the content *Dancer*. This is

mainly due to the specific characteristics of *Dancer* depth maps, which contains very large homogenous depth areas, where the concealment using a weighted interpolation of the undamaged neighbour values performs quite effectively (ECM1). For example, the PSNR of ECM1 can be higher than ECM2 up to 1.47dB in the content *Dancer*(QP=28, ER=40% using Error P. 2).

Considering the PSNR performance of the two methods using different QP, we can observe that the PSNR between ECM1 and ECM2 becomes closer to each other for higher QPs. This is due to the higher compression ratio, which has the effect of blurring the geometric details (edges/contours of shapes) of the depth maps. ECM2 strongly relies on this information in order to perform a more accurate concealment. The objective results obtained from this study are in line with the subjective results presented in Section 6.3.1. Both quality assessment methods show that for both *Book Arrival*, *Newspaper* and *Shark* ECM2 performs better than ECM1 while for *Dancer* ECM1 performs better than ECM2.

Analysing the methods performance at different error rates, we can observe in Table 6.4 that for Error P.1, when the error rate is higher, the PSNR difference between ECM2 and ECM1 is also higher. In the case of Error P.2, the error rate does not affect the PSNR difference between both concealment methods. These results are also in line with the subjective results shown in Figure 6.2(a).

When looking at the SSIM results, the information provided by this metric is not very consistent. The SSIM values are very close to each other and only for higher distortion in the synthesised images it is possible to observe a more clear SSIM difference.

A direct comparison between ECM2 and ECM1 was done, in order to find which ECM produces better virtual views according to each metric. Table 6.3 presents these results with answers to the following question "*Is ECM2 better than ECM1?*". When the answer is "1", it means that ECM2 is better than ECM1, when the answer is "0", it means the opposite. When the answer is "=", it means that both methods achieved the same result.

A direct comparison between the results of the objective metrics and the subjective tests was done, in order to find which method is more related to the subjective scores and to establish the preferences of users in regard to ECM1 and ECM2. Table 6.3 presents the relation between objective and subjective scores by performing and *XNOR* operation ($\odot$) between the results of whether ECM1 or ECM2 is perceived as having better quality

(SSIM$\odot$Subj.; PSNR$\odot$Subj; pPSNR$\odot$Subj). When the result is "1" it means that the objective metric produces the same result as subjective evaluation. When the answer is "0" it means that objective and subjective results are not related.

By observing the results shown in Table 6.3, the average ratio (in percentage) is computed by counting the number of times that each objective metric produces the same user preference as the subjective measure. It is possible to observe that pPSNR is more often matched with the subjective metric than PSNR and much more than SSIM. In the tested scenario, pPSNR yields the same preference as the subjective method in more than 66.67% of the tested cases. The minimum of 66.67% is achieved for *Dancer* sequence, due to the characteristics of its depth maps, which are similarly reconstructed by both methods. The *Overall ratio* presented at the bottom of Table 6.3 shows that the proposed pPSNR is able to match the subjective user preference in 87.5%, which is higher than the two other objective metrics. On the other hand, SSIM is the one that produces the worst results.

The work described in this chapter presents a study about quality assessment of reconstructed depth maps using different concealment algorithms to recover lost regions. The achieved results allow concluding that a perceptually-driven PSNR metric produces a high correspondence with the scores obtained through subjective evaluation. Therefore, it is reasonable to conclude that the proposed pPSNR can be used for evaluating the user preference of virtual views synthesised with recovered depth maps.

**Table 6.3** – Relation between objective and subjective metric.

| | Metric | SSIM⊙Subj. | | | PSNR⊙Subj. | | | pPSNR⊙Subj. | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Err. rate/QP | 28 | 34 | 40 | 28 | 34 | 40 | 28 | 34 | 40 |
| Seq. | Error Pattern 1 | | | | | | | | | |
| 1 | 5% | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 10% | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 20% | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| | 30% | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| | 40% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 50% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Av. Ratio(%) | 66.67 | | | 88.89 | | | 88.89 | | |
| 2 | 5% | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 10% | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| | 20% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 30% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 40% | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| | 50% | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Av. Ratio(%) | 55.56 | | | 88.89 | | | 88.89 | | |
| 3 | 5% | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 10% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 20% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 30% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 40% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 50% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Av. Ratio(%) | 88.89 | | | 100 | | | 100 | | |
| 4 | 5% | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| | 10% | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| | 20% | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 30% | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| | 40% | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| | 50% | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| | Av. Ratio(%) | 44.44 | | | 44.44 | | | 66.67 | | |
| | Error Pattern 2 | | | | | | | | | |
| 1 | 5% | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 20% | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| | 40% | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| | Av. Ratio(%) | 22.22 | | | 77.78 | | | 77.78 | | |
| 2 | 5% | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 20% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 40% | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| | Av. Ratio(%) | 44.44 | | | 88.89 | | | 88.89 | | |
| 3 | 5% | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 20% | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 40% | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Av. Ratio(%) | 55.55 | | | 100 | | | 100 | | |
| 4 | 5% | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| | 20% | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 40% | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Av. Ratio(%) | 55.56 | | | 88.89 | | | 88.89 | | |
| Overall Average ratio(%) | | 54.17 | | | 84.71 | | | 87.5 | | |

**Table 6.4** – Synthesised views objective evaluation using PSNR(dB), SSIM(%) and pPSNR(dB).

| | Metric | PSNR(dB) | | | | | | | | | SSIM(%) | | | | | | | | | pPSNR(dB) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Method | ECM 1 | | | ECM 2 | | | ECM2-ECM1 | | | ECM 1 | | | ECM2 | | | ECM2-ECM1 | | | ECM 1 | | | ECM2 | | | ECM2-ECM1 | | |
| | Err. Rate/QP | 28 | 34 | 40 | 28 | 34 | 40 | 28 | 34 | 40 | 28 | 34 | 40 | 28 | 34 | 40 | 28 | 34 | 40 | 28 | 34 | 40 | 28 | 34 | 40 | 28 | 34 | 40 |
| Seq. | Error Pattern 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Book Arrival | 0% | 40.07 | 37.78 | 35.10 | 40.07 | 37.78 | 35.10 | 0 | 0 | 0 | 95.7 | 94.1 | 91.5 | 95.7 | 94.1 | 91.5 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — |
| | 5% | 39.63 | 37.52 | 34.97 | 39.94 | 37.60 | 35.03 | 0.31 | 0.08 | 0.06 | 95.6 | 94.0 | 91.5 | 95.7 | 94.0 | 91.5 | 0.1 | 0 | 0 | 33.58 | 34.58 | 34.12 | 39.35 | 35.46 | 35.14 | 3.97 | 0.88 | 1.02 |
| | 10% | 39.32 | 37.35 | 34.89 | 39.79 | 37.70 | 35.08 | 0.47 | 0.35 | 0.19 | 95.6 | 94.0 | 91.4 | 95.6 | 94.0 | 91.5 | 0 | 0 | 0.1 | 32.88 | 33.56 | 33.75 | 36.38 | 38.27 | 37.68 | 3.50 | 4.71 | 3.92 |
| | 20% | 37.91 | 36.46 | 34.42 | 39.50 | 37.45 | 34.93 | 1.59 | 0.99 | 0.51 | 95.2 | 93.7 | 91.2 | 95.5 | 93.9 | 91.4 | 0.3 | 0.2 | 0.2 | 31.42 | 30.93 | 32.42 | 36.52 | 35.48 | 36.03 | 5.10 | 4.54 | 3.61 |
| | 30% | 37.26 | 36.02 | 34.19 | 38.22 | 36.15 | 34.35 | 0.96 | 0.13 | 0.16 | 95.0 | 93.5 | 91.1 | 95.3 | 93.7 | 91.2 | 0.3 | 0.2 | 0.1 | 30.78 | 30.36 | 30.36 | 32.82 | 30.68 | 30.89 | 2.04 | 0.32 | 0.53 |
| | 40% | 37.78 | 36.41 | 34.34 | 38.51 | 37.07 | 34.71 | 0.73 | 0.66 | 0.37 | 95.1 | 93.6 | 91.1 | 95.2 | 93.7 | 91.2 | 0.1 | 0.1 | 0.1 | 33.63 | 33.18 | 33.49 | 35.19 | 35.07 | 34.58 | 1.56 | 1.88 | 1.09 |
| | 50% | 37.10 | 35.82 | 34.02 | 37.95 | 36.37 | 34.48 | 0.85 | 0.55 | 0.46 | 95.0 | 93.4 | 91.0 | 95.1 | 93.6 | 91.2 | 0.1 | 0.2 | 0.2 | 32.82 | 32.38 | 32.92 | 34.21 | 33.65 | 33.68 | 1.49 | 1.27 | 0.76 |
| Newspaper | 0% | 37.57 | 35.40 | 33.16 | 37.57 | 35.40 | 33.16 | 0 | 0 | 0 | 96.4 | 94.5 | 91.4 | 96.4 | 94.5 | 91.4 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — |
| | 5% | 36.79 | 34.96 | 32.89 | 37.30 | 35.28 | 33.09 | 0.51 | 0.32 | 0.2 | 96.2 | 94.3 | 91.3 | 96.3 | 94.4 | 91.3 | 0.1 | 0.1 | 0 | 32.35 | 31.99 | 33.50 | 35.84 | 36.19 | 37.01 | 3.49 | 4.20 | 3.51 |
| | 10% | 36.70 | 34.88 | 32.86 | 36.85 | 35.07 | 32.92 | 0.15 | 0.19 | 0.06 | 96.2 | 94.3 | 91.3 | 96.2 | 94.3 | 91.3 | 0 | 0 | 0 | 32.05 | 31.97 | 32.48 | 32.59 | 33.69 | 33.25 | 0.54 | 1.72 | 0.77 |
| | 20% | 35.45 | 34.16 | 32.38 | 36.42 | 34.74 | 32.90 | 0.97 | 0.58 | 0.52 | 95.8 | 94.0 | 91.0 | 96.0 | 94.1 | 91.2 | 0.2 | 0.1 | 0.2 | 29.90 | 30.40 | 30.04 | 32.48 | 32.22 | 33.13 | 2.58 | 1.82 | 3.09 |
| | 30% | 34.77 | 33.63 | 32.04 | 35.59 | 34.10 | 32.51 | 0.82 | 0.47 | 0.47 | 95.6 | 93.8 | 90.9 | 95.8 | 94.0 | 91.1 | 0.2 | 0.2 | 0.2 | 30.26 | 30.53 | 30.14 | 32.18 | 31.96 | 32.34 | 1.92 | 1.43 | 2.20 |
| | 40% | 35.00 | 33.81 | 32.16 | 34.67 | 33.90 | 32.33 | -0.33 | 0.09 | 0.17 | 95.6 | 93.8 | 90.9 | 95.5 | 93.9 | 91.0 | -0.1 | 0.1 | 0.1 | 31.08 | 31.52 | 31.51 | 30.32 | 31.74 | 31.90 | -0.75 | 0.22 | 0.39 |
| | 50% | 34.52 | 33.43 | 31.91 | 34.86 | 33.55 | 32.36 | 0.34 | 0.12 | 0.45 | 95.3 | 93.6 | 90.7 | 95.3 | 93.6 | 90.9 | 0 | 0 | 0.2 | 31.36 | 31.51 | 31.36 | 31.96 | 31.75 | 32.73 | 0.60 | 0.24 | 1.37 |
| Shark | 0% | 41.13 | 37.90 | 35.03 | 41.13 | 37.90 | 35.03 | 0 | 0 | 0 | 96.8 | 93.7 | 88.8 | 96.8 | 93.7 | 88.8 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — |
| | 5% | 38.41 | 36.43 | 34.26 | 38.67 | 36.59 | 34.36 | 0.26 | 0.16 | 0.1 | 96.3 | 93.2 | 88.4 | 96.3 | 93.3 | 88.4 | 0 | 0.1 | 0 | 28.40 | 28.78 | 25.80 | 28.92 | 29.19 | 26.31 | 0.52 | 0.41 | 0.51 |
| | 10% | 37.84 | 36.08 | 34.04 | 38.64 | 36.60 | 34.35 | 0.8 | 0.52 | 0.31 | 96.2 | 93.2 | 88.4 | 96.3 | 93.3 | 88.5 | 0.1 | 0.1 | 0.1 | 26.96 | 27.16 | 25.75 | 28.53 | 28.89 | 27.56 | 1.57 | 1.73 | 1.81 |
| | 20% | 37.89 | 36.12 | 34.07 | 38.57 | 36.51 | 34.33 | 0.68 | 0.39 | 0.26 | 96.2 | 93.1 | 88.3 | 96.3 | 93.2 | 88.4 | 0.1 | 0.1 | 0.1 | 28.59 | 28.10 | 26.81 | 29.98 | 29.46 | 28.43 | 1.39 | 1.36 | 1.62 |
| | 30% | 37.12 | 35.57 | 33.75 | 38.25 | 36.35 | 34.24 | 1.13 | 0.78 | 0.49 | 96.1 | 93.0 | 88.2 | 96.2 | 93.1 | 88.4 | 0.1 | 0.1 | 0.2 | 28.87 | 27.87 | 27.03 | 30.74 | 29.85 | 29.20 | 1.92 | 1.97 | 2.18 |
| | 40% | 37.23 | 35.69 | 33.82 | 38.43 | 36.47 | 34.29 | 1.2 | 0.78 | 0.47 | 95.9 | 92.9 | 88.2 | 96.2 | 93.1 | 88.4 | 0.3 | 0.2 | 0.2 | 29.73 | 29.61 | 28.53 | 31.93 | 31.75 | 30.73 | 2.20 | 2.14 | 0.39 |
| | 50% | 36.66 | 35.23 | 33.57 | 38.20 | 36.41 | 34.28 | 1.54 | 1.18 | 0.71 | 95.8 | 92.8 | 88.1 | 96.1 | 93.1 | 88.3 | 0.3 | 0.3 | 0.2 | 29.68 | 29.42 | 28.27 | 32.22 | 32.11 | 31.12 | 2.54 | 2.69 | 2.85 |
| Dancer | 0% | 36.88 | 33.53 | 30.40 | 36.88 | 33.53 | 30.40 | 0 | 0 | 0 | 95.4 | 91.0 | 84.9 | 95.4 | 91.0 | 84.9 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — |
| | 5% | 35.92 | 33.09 | 30.18 | 36.31 | 33.20 | 30.31 | 0.39 | 0.11 | 0.13 | 95.2 | 90.9 | 84.8 | 95.3 | 90.9 | 84.9 | 0.1 | 0 | 0.1 | 29.51 | 28.60 | 27.47 | 31.47 | 29.18 | 30.37 | 1.96 | 0.58 | 2.90 |
| | 10% | 36.76 | 33.51 | 30.42 | 36.77 | 33.52 | 30.42 | 0.01 | 0.01 | 0 | 95.3 | 91.0 | 84.9 | 95.4 | 91.0 | 84.9 | 0.1 | 0 | 0 | 34.36 | 33.07 | 34.61 | 33.59 | 32.25 | 32.00 | -0.77 | -0.82 | -2.61 |
| | 20% | 34.85 | 32.62 | 30.05 | 33.85 | 31.84 | 29.70 | -1 | -0.78 | -0.35 | 94.8 | 91.0 | 84.6 | 95.0 | 90.6 | 84.6 | 0.2 | -0.4 | 0 | 29.69 | 28.20 | 27.66 | 27.75 | 26.06 | 25.24 | -1.94 | -2.14 | -2.42 |
| | 30% | 35.26 | 32.79 | 30.21 | 35.81 | 33.12 | 30.27 | 0.55 | 0.33 | 0.06 | 95.0 | 90.7 | 84.7 | 95.2 | 90.9 | 84.8 | 0.2 | 0.2 | 0.1 | 30.91 | 30.05 | 28.75 | 32.23 | 31.25 | 29.17 | 1.32 | 1.20 | 0.42 |
| | 40% | 34.08 | 32.24 | 29.89 | 33.18 | 31.45 | 29.49 | -0.9 | -0.79 | -0.4 | 94.5 | 90.4 | 84.4 | 94.8 | 90.5 | 84.5 | 0.3 | 0.1 | 0.1 | 28.43 | 27.66 | 27.37 | 26.96 | 26.01 | 25.21 | -1.48 | -1.65 | -2.15 |
| | 50% | 34.04 | 32.09 | 29.86 | 33.98 | 32.29 | 29.88 | -0.06 | 0.2 | 0.02 | 94.6 | 90.4 | 84.4 | 95.0 | 90.8 | 84.7 | 0.4 | 0.4 | 0.3 | 28.07 | 27.72 | 27.41 | 27.89 | 27.79 | 26.20 | -0.18 | 0.07 | -1.21 |
| | Error Pattern 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Book Arrival | 0% | 40.07 | 37.78 | 35.10 | 40.07 | 37.78 | 35.10 | 0 | 0 | 0 | 95.7 | 94.1 | 91.5 | 95.7 | 94.1 | 91.5 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — |
| | 5% | 39.79 | 37.63 | 35.02 | 39.89 | 37.66 | 35.05 | 0.1 | 0.03 | 0.03 | 95.6 | 94.0 | 91.5 | 95.6 | 94.0 | 91.5 | 0 | 0 | 0 | 37.36 | 37.66 | 37.87 | 39.10 | 37.77 | 38.39 | 1.74 | 0.11 | 0.52 |
| | 20% | 38.05 | 36.62 | 34.52 | 38.41 | 36.84 | 34.69 | 0.36 | 0.22 | 0.17 | 95.3 | 93.7 | 91.3 | 95.4 | 93.8 | 91.3 | 0.1 | 0.1 | 0 | 31.90 | 31.18 | 32.31 | 32.96 | 32.03 | 33.27 | 1.06 | 0.85 | 0.96 |
| | 40% | 37.68 | 36.35 | 34.34 | 38.53 | 36.90 | 34.70 | 0.85 | 0.55 | 0.36 | 95.2 | 93.7 | 91.2 | 95.2 | 93.7 | 91.2 | 0 | 0 | 0 | 31.79 | 31.81 | 32.43 | 33.17 | 32.85 | 33.70 | 1.38 | 1.03 | 1.28 |
| Newspaper | 0% | 37.57 | 35.40 | 33.16 | 37.57 | 35.40 | 33.16 | 0 | 0 | 0 | 96.4 | 94.5 | 91.4 | 96.4 | 94.5 | 91.4 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — |
| | 5% | 36.58 | 34.80 | 32.84 | 37.06 | 35.03 | 33.03 | 0.48 | 0.23 | 0.19 | 96.3 | 94.4 | 91.3 | 96.3 | 94.4 | 91.4 | 0 | 0 | 0.1 | 31.94 | 32.91 | 31.39 | 34.81 | 34.87 | 34.12 | 2.87 | 1.96 | 2.73 |
| | 20% | 35.95 | 34.46 | 32.62 | 36.16 | 34.65 | 32.70 | 0.21 | 0.19 | 0.08 | 96.0 | 94.2 | 91.2 | 96.0 | 94.2 | 91.2 | 0 | 0 | 0 | 32.07 | 31.89 | 32.08 | 32.13 | 32.59 | 32.30 | 0.06 | 0.70 | 0.22 |
| | 40% | 34.97 | 33.88 | 32.28 | 35.12 | 33.98 | 32.43 | 0.15 | 0.1 | 0.15 | 95.7 | 93.9 | 91.0 | 95.6 | 93.8 | 91.0 | -0.1 | -0.1 | 0 | 29.88 | 29.90 | 29.59 | 30.03 | 29.96 | 29.86 | 0.15 | 0.06 | 0.27 |
| Shark | 0% | 41.13 | 37.90 | 35.03 | 41.13 | 37.90 | 35.03 | 0 | 0 | 0 | 96.8 | 93.7 | 88.8 | 96.8 | 93.7 | 88.8 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — |
| | 5% | 38.40 | 36.41 | 34.24 | 38.56 | 36.55 | 34.28 | 0.16 | 0.14 | 0.04 | 96.3 | 93.2 | 88.4 | 96.3 | 93.2 | 88.4 | 0 | 0 | 0 | 27.42 | 26.75 | 25.91 | 24.42 | 26.75 | 26.54 | 0.38 | 0.55 | 0.63 |
| | 20% | 37.85 | 36.12 | 34.09 | 38.21 | 36.24 | 34.16 | 0.36 | 0.12 | 0.07 | 96.2 | 93.1 | 88.3 | 96.2 | 93.2 | 88.4 | 0 | 0.1 | 0.1 | 29.92 | 29.44 | 27.35 | 29.92 | 29.44 | 27.74 | 0.64 | 0.41 | 0.39 |
| | 40% | 36.79 | 35.41 | 33.68 | 37.46 | 35.97 | 33.98 | 0.67 | 0.56 | 0.3 | 95.9 | 92.9 | 88.1 | 96.0 | 93.0 | 88.3 | 0.1 | 0.1 | 0.2 | 31.33 | 30.98 | 28.10 | 31.33 | 30.98 | 25.33 | 1.08 | 1.44 | 1.23 |
| Dancer | 0% | 36.88 | 33.53 | 30.40 | 36.88 | 33.53 | 30.40 | 0 | 0 | 0 | 95.4 | 91.0 | 84.9 | 95.4 | 91.0 | 84.9 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — |
| | 5% | 36.72 | 33.46 | 30.38 | 36.17 | 33.16 | 30.30 | -0.55 | -0.3 | -0.08 | 95.4 | 91.0 | 84.9 | 95.3 | 91.0 | 84.9 | -0.1 | 0 | 0 | 33.51 | 31.48 | 30.06 | 29.05 | 26.08 | 24.41 | -4.46 | -5.40 | -5.65 |
| | 20% | 36.14 | 33.26 | 30.31 | 34.95 | 32.60 | 30.15 | -1.19 | -0.66 | -0.16 | 95.2 | 90.9 | 84.8 | 95.1 | 90.8 | 84.8 | -0.1 | -0.1 | 0 | 30.99 | 29.00 | 29.54 | 26.74 | 25.49 | 26.08 | -4.25 | -3.51 | -3.47 |
| | 40% | 35.71 | 33.01 | 30.25 | 34.24 | 32.25 | 29.90 | -1.47 | -0.76 | -0.35 | 95.1 | 90.8 | 84.8 | 95.0 | 90.8 | 84.7 | -0.1 | 0 | -0.1 | 30.90 | 29.15 | 28.82 | 27.54 | 25.88 | 24.73 | -3.36 | -3.27 | -4.09 |

# 7 Conclusions and future work

The investigation described in this thesis was carried out in the field of 3D video communications based on multivew video-plus-depth (MVD), which has been receiving a great deal of attention from the research community and related industry. The review of the current state of the art in chapters 2 and 3 highlighted the lack of efficient solutions to cope with transmission errors and packet loss in depth maps. This was the motivation for deeper research and development of novel methods for depth map error concealment with the objective of improving the quality of virtual views generated at the receivers.

In this thesis, it was found that the efficiency of spatial domain error concealment techniques highly depend on both the size of the lost regions and the characteristics of the depth map to be recovered. If a lost region is small, naturally, the uncorrupted neighbouring regions have much more closer values to the lost region, resulting in an easier recovery process. Regarding the characteristics of the image, when the lost regions contain high frequency content with periodic characteristics, the suitable methods for these cases are frequency domain algorithms. However, previous works have shown that depth map analysis in the frequency domain normally shows that such periodic characteristics do not exist with enough relevancy. When the missing depth map regions are spatially large, spatial domain EC techniques are difficult to recover and temporal and inter-view EC methods were found to perform better because in these cases there is higher correlation in the temporal and inter-view domain than in the spatially neighbouring regions. The novel error concealment methods developed in Chapters 4 and 5 were derived from these

findings.

Three new spatial domain error concealment methods were proposed in Chapter 4, based on the contour reconstruction of corrupted depth maps and spatial interpolation. These methods start by first pairing the broken segments, and then contour interpolation is done in two distinct ways: either using Bézier curves or using the corresponding texture images to interpolate the broken segments. When comparing the performance of these two methods, the average PSNR of the synthesised images are quite similar in the tested sequences, where their variation is not greater than 0.1dB. A variant method was devised, which recovers missing contour segments from both texture and Bézier interpolation. The results demonstrated that the combination of both methods leads to improved quality of the synthesised views up to 1.01 dB.

Further research on temporal domain techniques lead to the development of an error concealment method that takes into account not only the available data in the spatial domain but also inter-view similarities to recover missing regions (e.g. slices) in depth maps. Based on the disparity computed from texture images of adjacent views, a disparity compensation has proven that consistent PSNR gains can be achieved over other reference methods. This approach proved that exploiting inter-view correlation of texture images through disparity compensation is effective for reconstruction of lost depth.

Block matching based on geometric transformations (BMGT) was exploited in another depth reconstruction method, to take advantage of their ability to represent complex motion, such as rotations, zooming, etc. This was shown as particularly useful when using MVD sequences captured from non-linear camera arrangements. This method is able to efficiently recover lost regions in depth maps, even large block sizes such as $64 \times 64$, which were used in the experimental evaluation. The simulation results demonstrated that quite accurate reconstruction may be achieved, as PSNR gains of 5dB over the reference method were obtained. BMGT was also used to develop novel temporal error concealment methods for MVD, considering reliable motion vectors, characterised by null residual data in the corresponding texture images. This method exploits the similarities between motion information of texture images and corresponding depth maps. BMGT includes searching in both spatially, temporally and also in the adjacent view in order to exploit all types of correlations with possible geometric transformations between the data under comparison. This approach, used to recover missing depth maps data in MVD, was

successfully implemented and evaluated, achieving consistent quality gains over the best reference method. When looking at individual synthesised images, the PSNR advantage can be as high as twice the average PSNR in the case of video sequences.

Spatial and temporal domain error concealment methods were also investigated for multiple description decoders of depth streams. Uncorrupted spatially neighbouring information is combined with the coarsely decoded depth map provided by the available descriptions to improve the depth maps. Thus, the proposed method is able to enhance depth maps coarsely decoded from one single description when the other is lost, resulting in synthesised images with higher quality, up to 1.69dB for a PLR of 10%. Also MVs from the co-located texture regions of the corrupted depth map were also used to recover the coarsely decoded regions and by using residual information to efficiently select the most accurate MVs. The utilisation of motion information in the concealment process proved to be an efficient solution to enhance coarsely decoded depth map, up to 0.6dB when comparing with case with no motion information and by exploiting the available MVs, even whole frames can be recovered.

In the last chapter, a study concerning subjective quality evaluation of synthesised images using reconstructed depth maps was carried out to compare objective metrics and subjective opinions. In this study, a perceptually-driven objective quality metric was proposed (pPSNR), taking into account the most sensitive regions of the synthesised images with reconstructed depth maps, namely contours and object edges. To validate the proposed quality metric, subjective testing was performed, revealing high similarities between objective and subjective results that have an average value of 87.5%. The objective results provided by PSNR have also shown high similarity with the subjective results, but not as high as the proposed metric (pPSNR). Regarding SSIM, subjective testing led to the conclusion that SSIM is not a suitable metric to evaluate the quality of the synthesised images using depth maps reconstructed with error concealment methods because such metric does consistently relate to the subjective testing for the reason that the average similarity results are in average only 54.17%.

## 7.1 Future work

As suggestions for future investigation and considering the proposed EC techniques for depth maps described in this thesis, other EC techniques described in the literature for texture images could be used together with the proposed techniques that were specifically tailored for depth maps. This way, it would be possible to evaluate how errors in texture would affect the depth map error concealment performance, by assessing the quality of the corresponding synthesised images.

Error resilience techniques for depth maps are also a complementary solution for robust delivery system of depth maps in MVD. As described in detail throughout this thesis, depth map geometric information, namely contours are of major importance to achieve an accurate concealment performance. Depth contours could be transmitted as an additional layer in an SVC encoding scheme, in order to support the recovery process and preserving the depth map shapes. A study, to assess such approach could also be addressed, namely to evaluate the amount of additional information needed to transmit the additional layer.

Integration of EC algorithm in standard decoders is also a research issue that has no definitive solution in terms of complexity optimization. Since, EC implies an increase in decoding complexity, this should be minimised without compromising performance. Therefore, introducing computational complexity as an additional performance metric to take into account in depth map EC, brings a new set of parameters into the problem of optimising performance, which is not yet fully exploited.

Objective metrics for quality evaluation of 3D video is nowadays a very important research topic. In this thesis, this topic was addressed by developing an objective quality metric, specifically tailored to evaluate virtual views synthesised using reconstructed depth maps. As a future work, this study could be further developed, by performing an extensive subjective evaluation of synthesised images using depth maps recovered by a larger set EC techniques, in order to develop a universal and more accurate metric for evaluation of the impact of depth map artefacts in the quality of experience. Finally, a detailed study regarding objective quality assessment could also be performed with the goal of verifying the correlation of the existing metrics with the subjective testing, pointing out the directions to develop a general objective quality metric for synthesised video in MVD.

# Bibliography


[1] Y. Lu, Y. Liu, and S. Dey, "Optimizing cloud mobile 3D display gaming user experience by asymmetric object of interest rendering," in *Communications (ICC), 2015 IEEE International Conference on*, pp. 6842–6848, June 2015.

[2] O. Hugues, P. Fuchs, and O. Nannipieri, "New augmented reality taxonomy: Technologies and features of augmented environment," in *Handbook of Augmented Reality* (B. Furht, ed.), pp. 47–63, Springer New York, 2011.

[3] X. Zhang, Z. Fan, J. Wang, and H. Liao, "3D Augmented Reality Based Orthopaedic Interventions," in *Computational Radiology for Orthopaedic Interventions* (G. Zheng and S. Li, eds.), vol. 23 of *Lecture Notes in Computational Vision and Biomechanics*, pp. 71–90, Springer International Publishing, 2016.

[4] M. Tanimoto, "FTV: Free-viewpoint Television," *Signal Processing: Image Communication*, vol. 27, pp. 555 – 570, July 2012.

[5] J. Stuckler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3d modeling and tracking," *Journal of Visual Communication and Image Representation*, vol. 25, pp. 137 – 147, January 2014.

[6] Y. C. Y.-K. W. K. U. M. H. J. Lainema and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP J. ASP.*, vol. 2009, pp. 8:1–8:13, January 2008.

[7] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC Standard," in *Proceedings of the IEEE*, vol. 99, pp. 626 –642, April 2011.

[8] C. Hewage, S. Worrall, S. Dogan, S. Villette, and A. Kondoz, "Quality Evaluation of Color Plus Depth Map-Based Stereoscopic Video," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 3, pp. 304 –318, April 2009.

[9] C. Hewage and M. Martini, "Quality evaluation for real-time 3D video services," *2011 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1 –5, July 2011.

[10] S. Marcelino, P. Assuncao, S. Faria, and S. Soares, "Error recovery of image-based depth maps using Bézier curve fitting," in *International Conference on Image Processing (ICIP)*, (Brussels,Belgium), pp. 2293 – 2296, September 2011.

[11] S. Marcelino, P. Assuncao, S. Faria, and S. Soares, "Lost block reconstruction in depth maps using color image contours," in *Picture Coding Symposium (PCS)*, (Krakow, Poland), pp. 253 –256, May 2012.

[12] S. Marcelino, P. Assuncao, S. de Faria, and S. Soares, "Efficient depth error concealment for 3D video over error-prone channels," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2013 IEEE International Symposium on*, pp. 1–5, June 2013.

[13] S. M. Marcelino, P. Assuncao, S. Faria, and S. Soares, "Two-stage depth map error concealment using geometric fitting," in *Conf. on Telecommunications - ConfTele*, May 2013.

[14] S. Marcelino, P. Assuncao, S. Faria, and S. Soares, "Spatial error concealment for intra-coded depth maps in multiview video-plus-depth," *Submitted to Multimedia Tools and Applications*, 2015.

[15] S. Marcelino, P. Assuncao, S. Faria, L. Cruz, and S. Soares, "Slice loss concealment for depth maps in multiview-video plus depth," in *Conf. on Telecommunications - ConfTele*, Setpember 2015.

[16] S. Marcelino, P. Assuncao, S. Faria, and S. Soares, "Depth map concealment using interview warping vectors from geometric transforms," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pp. 1821–1825, September 2013.

[17] S. Marcelino, P. Assuncao, S. Faria, and S. Soares, "A method to recover lost depth data in multiview video-plus-depth communications," *Submited to Journal of Visual Communication and Image Representation, ELSEVIER*, pp. 1 – 12, 2015.

[18] P. Correia, S. Marcelino, P. Assuncao, S. Faria, S. Soares, C. Pagliari, and E. da Silva, "Enhancement method for multiple description decoding of depth maps subject to random loss," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*, pp. 1–4, IEEE, July 2014.

[19] S. Marcelino, P. Assuncao, S. Faria, and S. Soares, "Depth maps reconstruction in MDC with lost descriptions," in *Submitted to IEEE International BMSB 2016*, 2015.

[20] S. Marcelino, S. Faria, R. Pepion, P. L. Callet, P. Assuncao, and S. Soares, "Quality evaluation of depth map error concelment using a perceptually-aware objective metric," in *3DTV-Conference: Immersive and Interective 3D Media Experience over Networks (3DTV-CON), 2015*, pp. 1–4, IEEE, July 2015.

[21] N. Holliman, N. Dodgson, G. Favalora, and L. Pockett, "Three-dimensional displays: A review and applications analysis," *Broadcasting, IEEE Transactions on*, vol. 57, pp. 362–371, June 2011.

[22] I. Sexton and P. Surman, "Stereoscopic and autostereoscopic display systems," *Signal Processing Magazine, IEEE*, vol. 16, pp. 85–99, May 1999.

[23] F. L. Kooi and A. Toet, "Visual comfort of binocular and 3d displays," *Displays*, vol. 25, no. 23, pp. 99 – 108, 2004.

[24] P. Boher, T. Leroux, T. Bignon, and V. Collomb-Patton, "Multispectral polarization viewing angle analysis of circular polarized stereoscopic 3d displays," 2010.

[25] G. Lawton, "3d displays without glasses: coming to a screen near you," *Computer*, pp. 17–19, January 2011.

[26] G.-J. Lv, W.-X. Zhao, D.-H. Li, and Q.-H. Wang, "Polarizer parallax barrier 3d display with high brightness, resolution and low crosstalk," *Display Technology, Journal of*, vol. 10, pp. 120–124, February 2014.

[27] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, and Y. Suenaga, "Multipoint measuring system for video and sound - 100-camera and microphone system," in *Multimedia and Expo, 2006 IEEE International Conference on*, pp. 437–440, July 2006.

[28] K. Muller, P. Merkle, and T. Wiegand, "3-D Video Representation Using Depth Maps," in *Proceedings of the IEEE*, vol. 99, pp. 643 –656, April 2011.

[29] M. Tanimoto, T. Fujii, and K. Suzuki, "View synthesis algorithm in view synthesis reference software 3.5 (VSRS3.5) Document M16090, ISO/IEC JTC1/SC29/WG11 (MPEG)," May 2009.

[30] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "Multiview Imaging and 3DTV," *Signal Processing Magazine, IEEE*, vol. 24, pp. 10–21, November 2007.

[31] P. Kauff, N. Atzpadin, C. Fehn, M. Mller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Processing: Image Communication*, vol. 22, pp. 217 – 234, February 2007.

[32] S. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor - system description, issues and solutions," in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, pp. 35–35, June 2004.

[33] A. Smolic, "3D video and free viewpoint video from capture to display," *Pattern Recognition*, vol. 44, pp. 1958 – 1968, September 2011.

[34] I. Daribo and H. Saito, "A novel inpainting-based layered depth video for 3dtv," *Broadcasting, IEEE Transactions on*, vol. 57, pp. 533–541, June 2011.

[35] J. Shade, S. Gortler, L.-w. He, and R. Szeliski, "Layered depth images," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 231–242, ACM, September 1998.

[36] X. Cheng, L. Sun, and S. Yang, "Generation of layered depth images from multi-view video," in *Image Processing. ICIP, IEEE International Conference on*, vol. 5, pp. V–225, IEEE, September 2007.

[37] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Image Processing. ICIP, IEEE International Conference on*, vol. 1, pp. I – 201–I – 204, September 2007.

[38] B. Julesz, "Foundations of cyclopean perception.," 1971.

[39] F. Shao, G. Jiang, X. Wang, M. Yu, and K. Chen, "Stereoscopic video coding with asymmetric luminance and chrominance qualities," *IEEE Transactions on Consumer Electronics*, vol. 56, pp. 2460–2468, December 2010.

[40] J. Quan, M. Hannuksela, and H. Li, "Asymmetric spatial scalability in stereoscopic video coding," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pp. 1–4, May 2011.

[41] Y. Chen, S. Liu, Y.-K. Wang, M. M. Hannuksela, H. Li, and M. Gabbouj, "Low-complexity asymmetric multiview video coding," in *Multimedia and Expo, 2008 IEEE International Conference on*, pp. 773– –776, IEEE, April 2008.

[42] W. J. Tam, "Image and depth quality of asymmetrically coded stereoscopic video for 3D-TV," *JVT-W094, San Jose, CA*, April 2007.

[43] L. B. Stelmach, W. J. Tam, D. V. Meegan, A. Vincent, and P. Corriveau, "Human perception of mismatched stereoscopic 3D inputs," in *Image Processing. Proceedings. International Conference on*, vol. 1, pp. 5–8, IEEE, September 2000.

[44] A. Vetro, "Frame compatible formats for 3d video distribution," in *Image Processing (ICIP), 17th IEEE International Conference on*, pp. 2405–2408, September 2010.

[45] A. Vetro, S. Yea, and A. Smolic, "Toward a 3D video format for auto-stereoscopic displays," September 2008.

[46] "http://www.stereo3d.com/3dhome.htm, last accessed on 2015/09/03."

[47] G. Sullivan, J. Boyce, Y. Chen, J.-R. Ohm, C. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (HEVC)," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, pp. 1001–1016, December 2013.

[48] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, pp. 1461–1473, November 2007.

[49] Y. Su, A. Vetro, and A. Smolic, "Common test condition for multiview video coding, Hangzbou, China, Joint Video Team (JVT) Doc. JVT-U211," October 2006.

[50] Droese and C. Clemens, "Results of CE1-D on multiview video coding, montreux, switzerland, ISO/IEC JTC 1/SC 29/WG 11 (MPEG) Doc. M13247," April 2006.

[51] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Distributed multi-view video coding," January 2006.

[52] C. Fehn, P. Kauff, S. Cho, H. Kwon, N. Hur, and J. Kim, "Asymmetric coding of stereoscopic video for transmission over t-dmb," in *3DTV Conference, 2007*, pp. 1–4, May 2007.

[53] G. Saygili, C. Gurler, and A. Tekalp, "Quality assessment of asymmetric stereo video coding," in *Image Processing (ICIP), 17th IEEE International Conference on*, pp. 4009–4012, September 2010.

[54] C. Gurler, B. Gorkemli, G. Saygili, and A. Tekalp, "Flexible transport of 3-D video over networks," *Proceedings of the IEEE*, vol. 99, pp. 694–707, April 2011.

[55] F. Shao, G. Jiang, M. Yu, K. Chen, and Y.-S. Ho, "Asymmetric coding of multi-view video plus depth based 3-D video for view rendering," *Multimedia, IEEE Transactions on*, vol. 14, pp. 157–167, February 2012.

[56] Y. Chen and A. Vetro, "Next-generation 3D formats with depth map support," *MultiMedia, IEEE*, vol. 21, pp. 90–94, April 2014.

[57] H. Schwarz, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, K. Muller, H. Rhee, G. Tech, M. Winken, D. Marpe, and T. Wiegand, "Extension of high efficiency video coding (HEVC) for multiview video and depth data," in

*Image Processing (ICIP), 19th IEEE International Conference on*, pp. 205–208, September 2012.

[58] Y. Chen, T. Ikai, K. Kawamura, S. Shimizu, and T. Suzuki, "MV-HEVC and 3D-HEVC conformance draft 2, JCT-3V document JCT3V-J1008," tech. rep., 2014.

[59] S. Yea and A. Vetro, "View synthesis prediction for multiview video coding," *Signal Processing Image Communication*, vol. 24, pp. 89 – 100, January 2009.

[60] B. Micallef, C. Debono, and R. Farrugia, "Exploiting depth information for fast motion and disparity estimation in multi-view video coding," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pp. 1–4, May 2011.

[61] P. Assuncao, L. Pinto, and S. Faria, "3d media representation and coding," in *3D Future Internet Media* (A. Kondoz and T. Dagiuklas, eds.), pp. 9–38, Springer New York, 2014.

[62] S. M. Faria, C. J. Debono, P. Nunes, and N. M. Rodrigues, "3D Video Representation and Coding," in *Novel 3D Media Technologies* (A. Kondoz and T. Dagiuklas, eds.), pp. 25–48, Springer New York, October 2015.

[63] M. Hannuksela, Y. Chen, T. Suzuki, J. Ohm, and G. Sullivan, "3DAVC draft text 8, JCT-3V document JCT3V-F1002," tech. rep., 2013.

[64] Y. Chen, M. M. Hannuksela, T. Suzuki, and S. Hattori, "Overview of the MVC+D 3D video coding standard," *Journal of Visual Communication and Image Representation*, vol. 25, pp. 679 – 688, May 2014.

[65] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D High-Efficiency Video Coding for Multi-View Video and Depth Data," *Image Processing, IEEE Transactions on*, vol. 22, pp. 3366–3378, September 2013.

[66] "Joint collaborative team on 3D video coding extension development (JCT-3V) of ITU-T VCEG and ISO/IEC MPEG. common test conditions of 3DV core experiments. Technical report, JCT3V-B1100," tech. rep., October 2012.

[67] F. Jager, "Simplified depth map intra coding with an optional depth lookup table," in *3D Imaging (IC3D), 2012 International Conference on*, pp. 1–4, December 2012.

[68] F. Jager, "Contour-based segmentation and coding for depth map compression," in *Visual Communications and Image Processing (VCIP), IEEE*, pp. 1–4, November 2011.

[69] P. Merkle, C. Bartnik, K. Muller, D. Marpe, and T. Wiegand, "3D video: Depth coding based on inter-component prediction of block partitions," in *Picture Coding Symposium (PCS), 2012*, pp. 149–152, May 2012.

[70] G. Ballocca, P. D'Amato, M. Grangetto, and M. Lucenteforte, "Tile format: A novel frame compatible approach for 3D video broadcasting," in *Multimedia and Expo (ICME), IEEE International Conference on*, pp. 1–4, July 2011.

[71] M. Zamarin, M. Salmistraro, S. Forchhammer, and A. Ortega, "Edge-preserving intra depth coding based on context-coding and H.264/AVC," in *Multimedia and Expo (ICME), IEEE International Conference on*, pp. 1–6, July 2013.

[72] G. Bjøntegaard, "document VCEG-M33: Calculation of average PSNR differences between RD-curves," in *ITU-T VCEG Meeting, Austin, Texas, USA, Tech. Rep*, April 2001.

[73] G. Shen, W.-S. Kim, A. Ortega, J. Lee, and H. Wey, "Edge-aware intra prediction for depth-map coding," in *Image Processing (ICIP), 17th IEEE International Conference on*, pp. 3393–3396, September 2010.

[74] G. Shen, W.-S. Kim, S. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *Picture Coding Symposium (PCS)*, pp. 566–569, December 2010.

[75] B. T. Oh, H.-C. Wey, and D.-S. Park, "Plane segmentation based intra prediction for depth map coding," in *Picture Coding Symposium (PCS), 2012*, pp. 41–44, May 2012.

[76] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P. de With, and T. Wiegand, "The Effect of Depth Compression on Multiview Rendering Quality," *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008*, pp. 245 –248, May 2008.

[77] Y. Morvan, D. Farin, and P. de With, "Depth-image compression based on an r-d optimized quadtree decomposition for the transmission of multiview images," in *Image Processing, ICIP. IEEE International Conference on*, vol. 5, pp. V – 105–V – 108, September 2007.

[78] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *Information Theory, IEEE Transactions on*, vol. 35, pp. 299–315, March 1989.

[79] D. Graziosi, N. Rodrigues, C. Pagliari, E. da Silva, S. de Faria, M. Perez, and M. de Carvalho, "Multiscale recurrent pattern matching approach for depth map coding," in *Picture Coding Symposium (PCS)*, pp. 294–297, December 2010.

[80] L. Lucas, N. Rodrigues, C. Pagliari, E. da Silva, and S. de Faria, "Efficient depth map coding using linear residue approximation and a flexible prediction framework," in *Image Processing (ICIP), 19th IEEE International Conference on*, pp. 1305–1308, September 2012.

[81] L. Lucas, N. Rodrigues, C. Pagliari, E. da Silva, and S. de Farla, "Predictive depth map coding for efficient virtual view synthesis," in *Image Processing (ICIP) 20th IEEE International Conference on*, pp. 2058–2062, September 2013.

[82] N. Francisco, N. Rodrigues, E. da Silva, M. de Carvalho, S. de Faria, and V. Silva, "Scanned compound document encoding using multiscale recurrent patterns," *IEEE Transactions on Image Processing*, vol. 19, pp. 2712–2724, April 2010.

[83] "ISO/IEC JTC1/SC29/WG11, call for proposals on 3d video coding technology, Doc. N12036, Geneva, CH," tech. rep., March 2011.

[84] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map coding with distortion estimation of rendered view," in *IS&T/SPIE Electronic Imaging*, pp. 75430B–75430B, International Society for Optics and Photonics, January 2010.

[85] B. Oh, j. Lee, and D. Park, "Depth map coding based on synthesized view distortion function," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, pp. 1344–1352, November 2011.

[86] K. Muller, P. Merkle, G. Tech, and T. Wiegand, "3D video coding with depth modeling modes and view synthesis optimization," in *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pp. 1–4, December 2012.

[87] Y. Zhang, S. Kwong, S. Hu, and C.-C. Kuo, "Efficient multiview depth coding optimization based on allowable depth distortion in view synthesis," *Image Processing, IEEE Transactions on*, vol. 23, pp. 4879–4892, November 2014.

[88] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC Standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, pp. 1103–1120, September 2007.

[89] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, pp. 1649–1668, December 2012.

[90] J. Chen, J. Boyce, Y. Ye, and M. M. Hannuksela, "scalable high efficiency video coding draft 3, in joint collaborative team on video coding (JCT-VC) document JCTVC-N1008, 14th meeting: Vienna," tech. rep., 14, August 2013.

[91] Y. Wang, A. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proceedings of the IEEE*, vol. 93, pp. 57–70, January 2005.

[92] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, pp. 974–997, May 1998.

[93] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in H.264/AVC standard," *Journal of Visual Communication and Image Representation*, vol. 17, pp. 425 – 450, April 2006.

[94] A. Connie, P. Nasiopoulos, V. Leung, and Y. Fallah, "Video packetization techniques for enhancing H.264 video transmission over 3G networks," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pp. 800–804, January 2008.

[95] L.-W. Kang and J.-J. Leou, "An error resilient coding scheme for H.264/AVC video transmission based on data embedding," *Journal of Visual Communication and Image Representation*, vol. 16, pp. 93 – 114, February 2005.

[96] S. Wenger, "H.264/AVC over IP," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 645–656, July 2003.

[97] T. Stockhammer, M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 657–673, July 2003.

[98] S. Valente, C. Dufour, F. Groliere, and D. Snook, "An efficient error concealment implementation for MPEG-4 video streams," *Consumer Electronics, IEEE Transactions on*, vol. 47, pp. 568–578, August 2001.

[99] M. Kim, H. Lee, and S. Sull, "Spatial error concealment for H.264 using sequential directional interpolation," *Consumer Electronics, IEEE Transactions on*, vol. 54, pp. 1811–1818, November 2008.

[100] Z. Rongfu, Z. Yuanhua, and H. Xiaodong, "Content-adaptive spatial error concealment for video communication," *Consumer Electronics, IEEE Transactions on*, vol. 50, pp. 335–341, February 2004.

[101] Y. Xu and Y. Zhou, "H.264 video communication based refined error concealment schemes," *Consumer Electronics, IEEE Transactions on*, vol. 50, pp. 1135–1141, November 2004.

[102] Y.-K. Wang, M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," in *Image Processing, Proceedings International Conference on*, vol. 2, pp. II–729–II–732 vol.2, September 2002.

[103] J.-W. Suh and Y.-S. Ho, "Error concealment based on directional interpolation," *Consumer Electronics, IEEE Transactions on*, vol. 43, pp. 295–302, August 1997.

[104] S. Varadarajan and L. Karam, "An improved perception-based no-reference objective image sharpness metric using iterative edge refinement," in *Image Processing, ICIP. 15th IEEE International Conference on*, pp. 401–404, October 2008.

[105] W. Kwok and H. Sun, "Multi-directional interpolation for spatial error concealment," *Consumer Electronics, IEEE Transactions on*, vol. 39, pp. 455–460, June 1993.

[106] S.-C. Hsia, "An edge-oriented spatial interpolation for consecutive block error concealment," *Signal Processing Letters, IEEE*, vol. 11, pp. 577–580, June 2004.

[107] W. Kim, J. Koo, and J. Jeong, "Fine directional interpolation for spatial error concealment," *Consumer Electronics, IEEE Transactions on*, vol. 52, pp. 1050–1056, August 2006.

[108] D. Agrafiotis, D. R. Bull, and N. Canagarajah, "Spatial error concealment with edge related perceptual considerations," *Signal Processing: Image Communication*, vol. 21, pp. 130 – 142, February 2006.

[109] H. Gharavi and S. Gao, "Spatial interpolation algorithm for error concealment," in *Acoustics, Speech and Signal Processing, ICASSP . IEEE International Conference on*, pp. 1153–1156, March 2008.

[110] N. Kanopoulos, N. Vasanthavada, and R. Baker, "Design of an image edge detection filter using the sobel operator," *Solid-State Circuits, IEEE Journal of*, vol. 23, pp. 358–367, April 1988.

[111] W.-Y. Kung, C.-S. Kim, and C.-C. Kuo, "A spatial-domain error concealment method with edge recovery and selective directional interpolation," in *Acoustics, Speech, and Signal Processing. Proceedings ICASSP. IEEE International Conference on*, vol. 5, pp. V–700–3 vol.5, April 2003.

[112] J. Canny, "A Computational Approach to Edge Detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, pp. 679 –698, November 1986.

[113] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *International journal of image processing (IJIP)*, vol. 3, pp. 1–11, February 2009.

[114] Y. Zhao, D. Tian, M. M. Hannukasela, and M. Gabbouj, "Spatial error concealment based on directional decision and intra prediction," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pp. 2899–2902, IEEE, 2005.

[115] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *CSVT, IEEE Transactions on*, vol. 13, pp. 560 –576, July 2003.

[116] X. Lee, Y.-Q. Zhang, and A. Leon-Garcia, "Information loss recovery for block-based image coding techniques: a fuzzy logic approach," in *Visual Communications*, pp. 529–540, International Society for Optics and Photonics, October 1993.

[117] X. Lee, Y.-Q. Zhang, and A. Leon-Garcia, "Information loss recovery for block-based image coding techniques-a fuzzy logic approach," *Image Processing, IEEE Transactions on*, vol. 4, pp. 259–273, March 1995.

[118] K. Meisinger and A. Kaup, "Spatial error concealment of corrupted image data using frequency selective extrapolation," in *Acoustics, Speech, and Signal Processing. Proceedings. ICASSP. IEEE International Conference on*, vol. 3, pp. iii–209–12 vol.3, May 2004.

[119] K. Meisinger and A. Kaup, "Minimizing a weighted error criterion for spatial error concealment of missing image data," in *Image Processing, ICIP . International Conference on*, vol. 2, pp. 813–816 Vol.2, October 2004.

[120] A. Kaup, K. Meisinger, and T. Aach, "Frequency selective signal extrapolation with applications to error concealment in image communication," {*AEU*} - *International Journal of Electronics and Communications*, vol. 59, pp. 147 – 156, June 2005.

[121] A. Kaup and T. Aach, "Coding of segmented images using shape-independent basis functions," *ImageProcessing,IEEE Transactions on*, vol. 7, pp. 937–947, July 1998.

[122] S. Bandyopadhyay, Z. Wu, P. Pandit, and J. Boyce, "Frame loss error concealment for H. 264/AVC," *Doc. JVT-P072. Poznan, Poland*, 2005.

[123] S. Bandyopadhyay, Z. Wu, P. Pandit, and J. Boyce, "An error concealment scheme for entire frame losses for H.264/AVC," in *Sarnoff Symposium, IEEE*, pp. 1–4, March 2006.

[124] M. Al-Mualla, N. Canagarajah, and D. Bull, "Error concealment using motion field interpolation," in *Image Processing, ICIP. Proceedings. International Conference on*, pp. 512–516 vol.3, Octpber 1998.

[125] W.-M. Lam, A. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *Acoustics, Speech, and Signal Processing, ICASSP, 1993 IEEE International Conference on*, vol. 5, pp. 417–420 vol.5, April 1993.

[126] K.-W. Kang, S. H. Lee, and T. Kim, "Recovery of coded video sequences from channel errors," April 1995.

[127] Y. Xu and Y. Zhou, "Adaptive temporal error concealment scheme for h.264/avc video decoder," *Consumer Electronics, IEEE Transactions on*, vol. 54, pp. 1846–1851, November 2008.

[128] X. Chen, Y. Chung, and C. Bae, "Dynamic multi-mode switching error concealment algorithm for h.264/avc video applications," *Consumer Electronics, IEEE Transactions on*, vol. 54, pp. 154–162, February 2008.

[129] C.-C. Wang, C.-Y. Chuang, K.-R. Fu, and S. D. Lin, "An integrated temporal error concealment for h.264/avc based on spatial evaluation criteria," *Journal of Visual Communication and Image Representation*, vol. 22, no. 6, pp. 522 – 528, 2011.

[130] S. Garg and S. Merchant, "Interpolated candidate motion vectors for boundary matching error concealment technique in video," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 53, pp. 1039–1043, October 2006.

[131] W.-Y. Kung, C.-S. Kim, and C.-C. Kuo, "Spatial and temporal error concealment techniques for video transmission over noisy channels," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, pp. 789–803, July 2006.

[132] Y. Chen, Y. Hu, O. Au, H. Li, and C. W. Chen, "Video error concealment using spatio-temporal boundary matching and partial differential equation," *Multimedia, IEEE Transactions on*, vol. 10, pp. 2–15, Jan 2008.

[133] T. Thaipanich, P.-H. Wu, and C.-C. Kuo, "Low-complexity video error concealment for mobile applications using obma," *Consumer Electronics, IEEE Transactions on*, vol. 54, pp. 753–761, May 2008.

[134] Q. Peng, T. Yang, and C. Zhu, "Block-based temporal error concealment for video packet using motion vector extrapolation," in *Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on*, vol. 1, pp. 10–14 vol.1, June 2002.

[135] S. Belfiore, M. Grangetto, E. Magli, and G. Olmo, "An error concealment algorithm for streaming video," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 3, pp. III–649–52 vol.2, September 2003.

[136] P. Salama, N. Shroff, and E. Delp, "Error concealment in mpeg video streams over atm networks," *Selected Areas in Communications, IEEE Journal on*, vol. 18, pp. 1129–1144, June 2000.

[137] Y. Chen, K. Yu, J. Li, and S. Li, "An error concealment algorithm for entire frame loss in video transmission," in *Picture Coding Symposium*, pp. 15–17, Citeseer, 2004.

[138] B. Yan and H. Gharavi, "A hybrid frame concealment algorithm for h.264/avc," *Image Processing, IEEE Transactions on*, vol. 19, pp. 98–107, Jan 2010.

[139] H. Liu, D. Wang, W. Li, and O. Issa, "New method for concealing entirely lost frames in h.264 video transmission over wireless networks," in *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on*, pp. 112–116, June 2011.

[140] K. Song, T. Chung, Y. Kim, Y. Oh, and C.-S. Kim, "Error concealment of h.264/avc video frames for mobile video broadcasting," *Consumer Electronics, IEEE Transactions on*, vol. 53, pp. 704–711, May 2007.

[141] S. Liu, Y. Chen, Y.-K. Wang, M. Gabbouj, M. Hannuksela, and H. Li, "Frame loss error concealment for multiview video coding," in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pp. 3470 –3473, May 2008.

[142] Y. Chen, C. Cai, and K.-K. Ma, "Stereoscopic video error concealment for missing frame recovery using disparity-based frame difference projection," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 4289–4292, November 2009.

[143] R. C. Gonzalez, *Digital image processing*. Pearson Education India, 2007.

[144] C. Bilen, A. Aksay, and G. Akar, "Motion and disparity aided stereoscopic full frame loss concealment method," in *Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th*, pp. 1–4, June 2007.

[145] L. Pang, M. Yu, W. Yi, G. Jiang, W. Liu, and Z. Jiang, "Relativity analysis-based error concealment algorithm for entire frame loss of stereo video," in *Signal Processing, 2006 8th International Conference on*, vol. 2, pp. –, 2006.

[146] T.-Y. Chung, S. Sull, and C.-S. Kim, "Frame loss concealment for stereoscopic video based on inter-view similarity of motion and intensity difference," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 441 –444, September 2010.

[147] B. Micallef and C. Debono, "Error concealment techniques for multi-view video," in *Wireless Days (WD), 2010 IFIP*, pp. 1 –5, October. 2010.

[148] B. Micallef, C. Debono, and R. Farrugia, "Performance of enhanced error concealment techniques in multi-view video coding systems," in *Systems, Signals and Image Processing (IWSSIP), 2011 18th International Conference on*, pp. 1 –4, June 2011.

[149] O. Stankiewicz, K. Wegner, and M. Doman andski, "Error concealment for MVC and 3D video coding," in *Picture Coding Symposium (PCS)*, (Nagoya, Japan), pp. 498 –501, December 2010.

[150] X. Xiang, D. Zhao, S. Ma, and W. Gao, "Auto-regressive model based error concealment scheme for stereoscopic video coding," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 849 –852, May 2011.

[151] X. Wu, K. Barthel, and W. Zhang, "Piecewise 2d autoregression for predictive image coding," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pp. 901–904 vol.3, Oct 1998.

[152] A. Ali, H. Karim, N. Arif, and A. Sali, "Depth image-based spatial error concealment for 3-D video transmission," in *Research and Development (SCOReD), 2010 IEEE Student Conference on*, pp. 421 –425, December 2010.

[153] B. Yan and J. Zhou, "Efficient Frame Concealment for Depth Image Based 3D Video Transmission," *Multimedia, IEEE Transactions on*, vol. PP, p. 1, June 2012.

[154] V.-H. Doan, V.-A. Nguyen, and M. Do, "Efficient view synthesis based error concealment method for multiview video plus depth," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pp. 2900–2903, May 2013.

[155] Y. Liu, J. Wang, and H. Zhang, "Depth Image-Based Temporal Error Concealment for 3-D Video Transmission," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, pp. 600 –604, April 2010.

[156] C. Hewage, S. Worrall, S. Dogan, and A. Kondoz, "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*, (Istanbul, Turkey), pp. 149 –152, May 2008.

[157] C. Hewage and M. Martini, "Joint Error Concealment Method for Backward Compatible 3D Video Transmission," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1 –5, May 2011.

[158] B. Yan, "A Novel H.264 Based Motion Vector Recovery Method for 3D Video Transmission," *Consumer Electronics, IEEE Transactions on*, vol. 53, pp. 1546 –1552, November 2007.

[159] C. Hewage, S. Worrall, S. Dogan, and A. Kondoz, "Frame concealment algorithm for stereoscopic video using motion vector sharing," in *IEEE International Conference on Multimedia and Expo*, pp. 485 –488, April 2008.

[160] T. Chung, S. Sull, and C. Kim, "Frame loss concealment for stereoscopic video plus depth sequences," *Consumer Electronics, IEEE Transactions on*, vol. 57, pp. 1336 –1344, August 2011.

[161] C. Bilen, A. Aksay, and G. B. Akar, "Two novel methods for full frame loss concealment in stereo video," in *Proc. Picture Coding Symposium*, Citeseer, November 2007.

[162] X. Liu, Q. Peng, and X. Fan, "Frame loss concealment for multi-view video plus depth," in *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on*, pp. 208 –211, June 2011.

[163] C. Hewage and M. Martini, "Joint Error Concealment Method for Backward Compatible 3D Video Transmission," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1 –5, May 2011.

[164] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video processing and communications*, vol. 5. Prentice Hall Upper Saddle River, 2002.

[165] W.-N. Lie and G.-H. Lin, "Error concealment for 3d video transmission," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pp. 2856–2559, May 2013.

[166] X. Zhang, Y. Zhao, C. Lin, H. Bai, C. Yao, and A. Wang, "Warping-driven mode selection for depth error concealment," in *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pp. 302–306, Dec 2014.

[167] J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, J. D. Foley, S. K. Feiner, and K. Akeley, *Computer Graphics, Principles and Practice*. Addison Wesley, 1990.

[168] R. Hasimoto-Beltran and A. Khokhar, "Spatial error concealment based on bezier curves," in *IEEE ICME 2005.*, pp. 996 –999, 2005.

[169] L. Soares and F. Pereira, "Spatial shape error concealment for object-based image and video coding," *Image Processing, IEEE Transactions on*, vol. 13, pp. 586 –599, April 2004.

[170] M.-J. Chen, C.-C. Cho, and M.-C. Chi, "Spatial and temporal error concealment algorithms of shape information for MPEG-4 video," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, pp. 778 – 783, June 2005.

[171] L. Atzori and F. G. Natale, "Reconstruction of missing or occluded contour segments using bezier interpolations," *Signal Processing*, vol. 80, pp. 1691 – 1694, August 2000.

[172] F. H. Foley, Andries van Dam, *Computer Gaphics, Principles and Practice*. Addison Wesley, 1990.

[173] http://iphome.hhi.de/suehring/tml/, *Suehring JM17H.264/AVC, last accessed on 2013/01/07*.

[174] L. Do, S. Zinger, and P. de With, "Objective quality analysis for free-viewpoint DIBR," in *ICIP, 2010*, pp. 2629 –2632, September 2010.

[175] Q. Xu, C. Chakrabarti, and L. Karam, "A distributed Canny edge detector and its implementation on FPGA," pp. 500 –505, Jan. 2011.

[176] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, pp. 600 –612, April 2004.

[177] http://opencv.itseez.com/, *OpenCV-2.1.0, Last accessed on 02/06/2014*.

[178] S. Kosov, T. Thormhlen, and H.-P. Seidel, "Accurate real-time disparity estimation with variational methods," vol. 5875 of *Lecture Notes in Computer Science*, pp. 796–807, Springer Berlin / Heidelberg, 2009.

[179] http://vision.middlebury.edu/stereo/eval/, *Middlebury Stereo Evaluation, Last accessed on 02/06/2014*.

[180] R. Jain, K. K. Ramakrishnan, and D. M. Chiu, "Definition of a general and intuitive loss model for packet networks and its implementation in the netem module in the linux kernel," tech. rep., Digital Equipment Corporation, September 2010.

[181] M. Ghanbari, S. de Faria, I. Goh, and K. Tan, "Motion compensation for very low bit-rate video," *Signal Processing: Image Communication*, vol. 7, pp. 567 – 580, 1995.

[182] P. Yin, H.-Y. Tourapis, A. Tourapis, and J. Boyce, "Fast mode decision and motion estimation for JVT/H.264," in *Proceedings of IEEE International Conference on Image Processing 2003, ICIP 2003.*, vol. 3, pp. III–853–6 vol.2, September 2003.

[183] P. Frossard, J. De Martin, and M. Civanlar, "Media streaming with network diversity," *Proceedings of the IEEE*, vol. 96, pp. 39–53, January 2008.

[184] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proceedings of the IEEE*, vol. 93, pp. 57–70, January 2005.

[185] P. Correia, P. Assuncao, and V. Silva, "Multiple description of coded video for path diversity streaming adaptation," *IEEE Transactions on Multimedia*, vol. 14, pp. 923–935, June 2012.

[186] A. Norkin, A. Gotchev, K. Egiazarian, and J. Astola, "Two stage multiple description image coders: Analysis and comparative study," *Signal Processing: Image Communications*, vol. 11, pp. 609–625, April 2006.

[187] H. A. Karim, A. Sali, S. Worrall, A. H. Sadka, and A. Kondoz, "Multiple description video coding for stereoscopic 3D," *Consumer Electronics, IEEE Transactions on*, vol. 55, pp. 2048–2056, November 2009.

[188] M.B.Dissanayake, D. D. Silva, S.T.Worral, and W. Fernando, "Error resilience technique for multi-view coding using redundant disparity vectors," in *IEEE International Conference On Multimedia and Expo, ICME, 2010*, pp. 1712–1717, July 2010.

[189] E. Ekmekcioglu, B. Gnel, M. Dissanayake, S. T. Worral, and A. M. Kondoz, "A scalable multi-view audiovisual entertainment framework with content-aware distribuition," in *IEEE International Conference On Image Processing, ICIP, 2010*, pp. 2401–2404, September 2010.

[190] Z. Liu, G. Cheung, J. Chakareski, and Y. Ji, "Multiple description coding of free viewpoint video for multi-path network streaming," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 2150–2155, December 2012.

[191] X. Wang and C. Cai, "Mode duplication based multiview multiple description video coding," in *Data Compression Conference (DCC)*, pp. 527–527, March 2013.

[192] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Transactions On Information Theory*, vol. 39, pp. 821–834, May 1993.

[193] E. Bosc, R. Pepion, P. Le Callet, M. Pressigout, and L. Morin, "Reliability of 2D quality assessment methods for synthesized views evaluation in stereoscopic viewing conditions," in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, pp. 1–4, October 2012.

[194] E. Bosc, R. Pepion, P. Le Callet, M. Koppel, P. Ndjiki-Nya, M. Pressigout, and L. Morin, "Towards a new quality metric for 3-D synthesized view assessment," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, pp. 1332–1343, November 2011.

[195] E. Bosc, P. Hanhart, P. Le Callet, and T. Ebrahimi, "A quality assessment protocol for free-viewpoint video sequences synthesized from decompressed depth data," in *Quality of Multimedia Experience (QoMEX), 2013 Fifth International Workshop on*, pp. 100–105, July 2013.

[196] P. Hanhart, E. Bosc, P. L. Callet, and T. Ebrahimi, "Free-viewpoint video sequences: A new challenge for objective quality metrics," in *Multimedia Signal Processing (MMSP), 2014 IEEE 16th International Workshop on*, pp. 1–6, September 2014.

[197] H. J. Seltman, *Experimental Design and Analysis, http://www.stat.cmu.edu/ hseltman/309/Book/Book.pdf, last accessed on 2014/12/10.* November 2014.

# Test sequences

This appendix describes the multiview video-plus-depth test sequences used for error concealment tests. All test sequences were in an uncompressed 4:2:0 YUV format with eight bits per sample. For each sequence, a printout example of the texture image and the corresponding depth map is shown.

## Ballet, Figure A.1

- Number of views: 8;
- Number of frames: 100;
- Frame rate: 15Hz;
- Resolution: 1024×768;
- Authors: C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, High-quality video view interpolation using a layered representation, ACM Trans. Graph., vol. 23, no. 3, Aug. 2004, pp. 600608.

## Breakdancers, Figure A.2

- Number of views: 8;
- Number of frames: 100;
- Frame rate: 15Hz;

- Resolution: 1024×768;
- Authors: C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, High-quality video view interpolation using a layered representation, ACM Trans. Graph., vol. 23, no. 3, Aug. 2004, pp. 600608.

## Beergarden, Figure A.3

- Number of views: 2;
- Number of frames: 150;
- Frame rate: 25Hz;
- Resolution: 1920×1080;
- Author: Philips® research.

## Kendo, Figure A.4

- Number of views: 7;
- Number of frames: 400;
- Frame rate: 30Hz;
- Resolution: 1024×768;
- Author: Nagoya university sequence, provided by Fujii Laboratory at Nagoya University.

## Balloons, Figure A.5

- Number of views: 7;
- Number of frames: 400;
- Frame rate: 30Hz;
- Resolution: 1024×768;
- Author: Nagoya university sequence, provided by Fujii Laboratory at Nagoya University.

## Book Arrival, Figure A.6

- Number of views: 16;
- Number of frames: 100;
- Frame rate: 25Hz;
- Resolution: 1024×768;
- Author: Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, HHI.

## Champagne, Figure A.7

- Number of views: 80;
- Number of frames: 300;
- Frame rate: 30Hz;
- Resolution: 1280×960;
- Author: Nagoya university sequence, provided by Fujii Laboratory at Nagoya University.

## Dancer, Figure A.8

- Number of views: 9;
- Number of frames: 250;
- Frame rate: 30Hz;
- Resolution: 1920×1080;
- Author: Nokia Reasearch.

## Shark, Figure A.9

- Number of views: 9;
- Number of frames: 300;
- Frame rate: 25Hz;

- Resolution: 1920×1088;
- Author: National Institute of Information and Communications Technology (NICT), Japan.

## Newspaper, Figure A.10

- Number of views: 8;
- Number of frames: 100;
- Frame rate: 25Hz;
- Resolution: 1024×768;
- Author: ISO/IEC JTC1/SC29/WG11, Multiview video test sequence and camera parameters, Document M15419, ISO, Archamps, France, 2008.

(a) Texture image. (b) Depth map.

**Figure A.1** – First frame of *Ballet* sequence (view 0).

(a) Texture image. (b) Depth map.

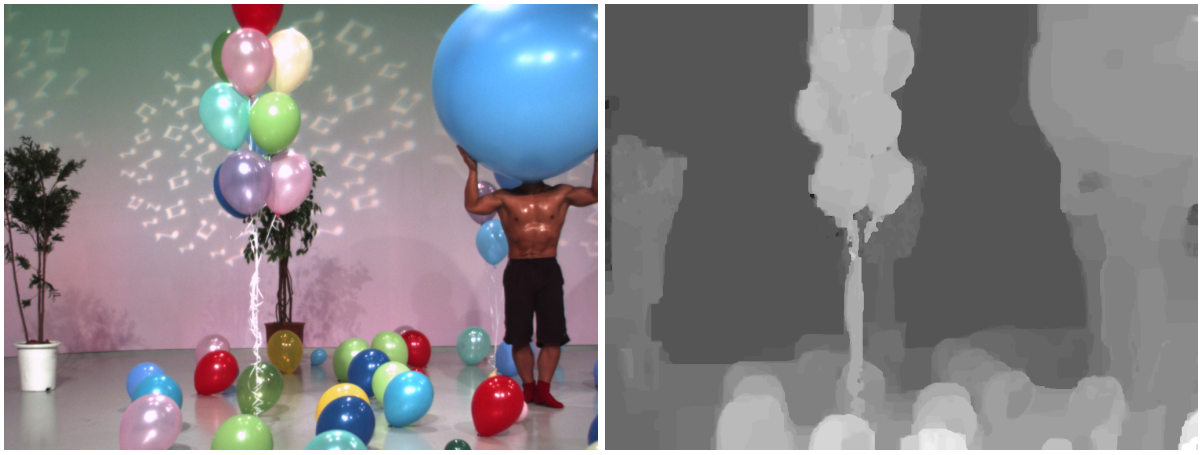
**Figure A.2** – First frame of *Breakdancers* sequence (view 0).

(a) Texture image. (b) Depth map.

**Figure A.3** – First frame of *Beergarden* sequence (view 1).

(a) Texture image. (b) Depth map.

**Figure A.4** – First frame of *Kendo* sequence (view 0).



(a) Texture image. (b) Depth map.

**Figure A.5** – First frame of *Balloons* sequence (view 1).

(a) Texture image. (b) Depth map.

**Figure A.6** – First frame of *Book Arrival* sequence (view 8).

(a) Texture image. (b) Depth map.

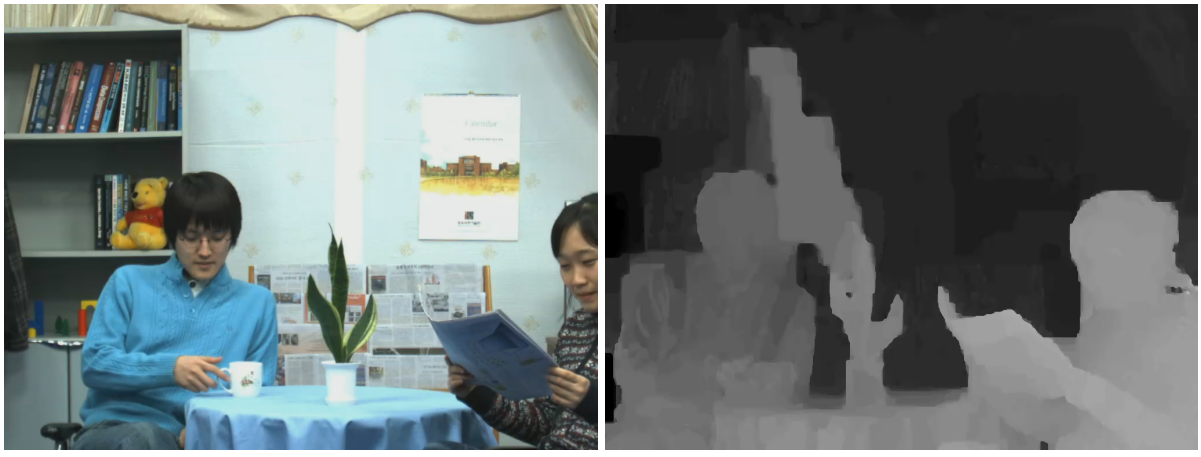
**Figure A.7** – First frame of *Champagne* sequence (view 39).

(a) Texture image. (b) Depth map.

**Figure A.8** – First frame of *Dancer* sequence (view 1).

(a) Texture image. (b) Depth map.

**Figure A.9** – First frame of *Shark* sequence (view 1).



(a) Texture image. (b) Depth map.

**Figure A.10** – First frame of *Newspaper* sequence (view 1).