Full Reference Quality Assessment of DIBR-based Synthesized Images

Miguel António Perdigão de Oliveira

Thesis to obtain the Master of Science Degree in Electrical and Computer Engineering

Supervisor: Prof. Maria Paula Dos Santos Queluz Rodrigues
Prof. João Miguel Duarte Ascenso

Examination Committee
Chairperson: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino
Supervisor: Prof. Maria Paula Dos Santos Queluz Rodrigues
Members of the Committee: Prof. António Manuel Gonçalves Pinheiro

November 2017
Acknowledgments

In first place, I would like to thank my parents, José António and Maria José, my sister Catarina and my brother Ricardo for always being there for me and for all the support. I also want to give a special word to Marta for sharing this journey with me.

I also would like to thank Prof. Paula Queluz and Prof. João Ascenso for all the patient and for always pushing for me. I truly believe that now I am a better professional then when I started working in this Thesis. I will certainly keep all the valuable lessons that I learned, during this past months, in my professional life, especially the importance of having a critical spirit when facing results.

Finally, but not the least, I would like to thank all my friends for all the good times and also to my colleagues from the Lab 10.16, who I had the pleasure to share these past months and for all the mutual help, namely: Fábio Rodrigues, Luís Nunes, Pedro Perdigão, Frederico Rodrigues, David Dias and Guilherme Moreira.
Abstract

Nowadays, 3D video applications and services are emerging, as a result of several technological advances that have led to a significant reduction in the cost of 3D acquisition devices and 3D displays. New visualization experiences, such as Free Viewpoint Video (FVV) are now possible providing more immersive, interactive and appealing experiences to the end user, than those provided by the classic 2D video. FVV enables the user to interactively control the viewpoint of a scene, without disruptions in the displayed video while changing the different viewpoints. However, due to bandwidth constraints, only a few views are transmitted and the rest are synthesized in the receiver side. The synthesis of intermediate viewpoints is usually made with a Depth Image Based Rendering (DIBR) technique, which requires the transmission of additional depth data. Thus, several artifacts may occur in the synthesized views, due to the lack of accuracy in the depth acquisition, compression or transmission errors in the depth maps, and errors due to the view synthesis solution (especially relevant in occluded areas). For the full acceptance of FVV, a high users’ QoE should be guaranteed. In order to allow an accurate estimation of the users’ perceived quality, this dissertation proposes a new quality assessment metric (full-reference) for synthesized images. The main goal of this metric is to detect and quantify the synthesis artifacts that most affect the QoE (e.g., edges distortions). The metric was evaluated using two distinct databases, and compared with conventional 2D Quality Metrics, as well as with metrics specifically addressing the quality assessment of synthesized images. The results show that the proposed metric provides a good prediction of the image quality perceived by the users, outperforming the considered state-of-the-art metrics.

Key-words: 3D video, FVV, MVD, Quality Assessment, Image Synthesis, Quality of Experience.
Resumo

Hoje em dia, as aplicações e serviços de vídeo 3D atravessam um momento de expansão e afirmação, resultante dos diversos avanços tecnológicos que conduziram a uma redução significativa do custo de aquisição de dispositivos 3D. Novas experiências de visualização, como é o caso do Free Viewpoint Video (FVV), são agora possíveis, permitindo experiências mais imersivas, interactivas e apelativas para o utilizador, comparativamente com o clássico vídeo 2D. A aplicação FVV permite que o utilizador, de forma interactiva, controle o ponto de vista ou perspectiva da cena, sem disrupções na exibição do vídeo nas mudanças de perspectiva. Contudo, devido aos limites de banda, apenas algumas perspectivas (ou "vistas") podem ser transmitidas, sendo as restantes sintetizadas no receptor. A síntese de vistas intermédias é tipicamente feita recorrendo a uma técnica designada por Depth Image Based Rendering (DIBR). No entanto, esta técnica pode conduzir ao aparecimento de artefactos nas vistas sintetizadas, devido à falta de precisão dos mapas de profundidade, a erros de compressão ou de transmissão dos respectivos mapas e a limitações da técnica de síntese utilizada (especialmente relevantes em áreas onde existem oclusões). Para uma total aceitação do FVV, é necessário garantir uma boa qualidade de experiência aos utilizadores. De forma a garantir uma correcta previsão da qualidade apercebida pelos utilizadores, nesta dissertação propõe-se uma nova métrica de avaliação de qualidade para as vistas sintetizadas. O objectivo desta nova métrica é detectar e quantificar os artefactos de síntese com maior impacto na qualidade de experiência dos utilizadores (e.g., distorções nos contornos dos objectos). O desempenho da métrica foi avaliado usando duas bases de dados distintas, e comparado com o desempenho de métricas de qualidade 2D convencionais, bem como com o desempenho de métricas desenvolvidas especificamente para a avaliação de qualidade de imagens sintetizadas. Os resultados mostram que a métrica proposta fornece uma boa previsão da qualidade de imagem apercebida pelos utilizadores, superando as métricas consideradas como representativas do estado-da-arte.

Palavras-chave: Vídeo 3D, FVV, Avaliação de Qualidade, Imagem sintetizada, Qualidade de Experiência.
# Table of Contents

Acknowledgments...........................................................................................................i
Abstract............................................................................................................................iii
Resumo..............................................................................................................................v
List of Figures..................................................................................................................xi
List of Tables..................................................................................................................xv
Acronyms........................................................................................................................xvi

Chapter 1. Introduction......................................................................................................1
  1.1. Context and Motivation .......................................................................................1
  1.2. Objectives .............................................................................................................2
  1.3. Main Contributions ..............................................................................................2
  1.4. Thesis Outline ......................................................................................................3

Chapter 2. 3D Video Transmission Chain ......................................................................5
  2.1 Introduction ............................................................................................................5
  2.2 Basic Concepts .......................................................................................................5
  2.3 3D Video Representation .......................................................................................8
  2.4 3D Video Content Compression ...........................................................................13
    2.4.1 Coding Standards for 2D Video ..................................................................13
    2.4.2 Simulcast Coding for 3D Video .................................................................14
    2.4.3 Frame-compatible Stereo Interleaving.......................................................14
    2.4.4 Texture Based Coding Standards for 3D Video .......................................15
    2.4.5 Texture-plus-depth Based Coding Standards for 3D Video ...................16
  2.5 3D Video Rendering .............................................................................................17
    2.5.1 Common Applications .............................................................................17
    2.5.2 Common Approaches for 3D Video Rendering .......................................17
  2.6 3D Video Capture and Display Technologies ....................................................19
    2.6.1 3D Video Capture ....................................................................................19
    2.6.2 3D Video Displays ....................................................................................20

Chapter 3. Objective Quality Assessment of 3D Synthesized Videos .............................23
  3.1 Introduction ............................................................................................................23
  3.2 Perspective Projection .........................................................................................23
3.3 View Synthesis Through Depth-Image Based Rendering ........................................ 24
  3.3.1 Depth Map Projection with 3D Warping ..................................................... 25
  3.3.2 Post-filtering on the Projected Depth Maps ............................................. 25
  3.3.3 Rendering Using Boundary Matting and Inpainting ................................ 26
3.4 Artifacts Resulting From Depth-Image Based View Synthesis .......................... 28
3.5 Quality Assessment Models .............................................................................. 29
  3.5.1 Common 2D Metrics .................................................................................. 31
  3.5.2 Depth Based Perceptual Quality Assessment ........................................... 32
  3.5.3 Quality Assessment Based on Spatial Features ....................................... 34
  3.5.4 Quality Assessment Based on Contours Consistency and Critical Areas ....... 36
  3.5.5 Quality Assessment Based on SSIM and Non-consistent Edges Shifts ......... 37
  3.5.6 Quality Assessment Based on Skin Detection ....................................... 39
  3.5.7 Perceptual Quality Assessment of 3D Synthesized Images ....................... 41

Chapter 4. Proposed Models for Quality Assessment of Synthesized Images ............... 43
  4.1 Introduction ................................................................................................. 43
  4.2 Approach ................................................................................................. 43
  4.3 2D Image Quality Metric .......................................................................... 44
  4.4 Structural Quality Metric ......................................................................... 47
    4.4.1 Hausdorff Distance Based Approach ................................................. 47
    4.4.2 Gradient Based Approach .................................................................. 53
  4.5 Fusion ........................................................................................................ 56

Chapter 5. Performance Evaluation ........................................................................ 59
  5.1 Introduction ............................................................................................... 59
  5.2 Test Material ............................................................................................ 59
    5.2.1 IRCCyN Database ............................................................................... 59
    5.2.2 SIAT-based Dataset ........................................................................... 62
  5.3 Logistic Function and Metrics Performance Measurement ........................... 63
  5.4 Performance Assessment Methodology ...................................................... 65
  5.5 Results and Analysis .................................................................................. 66
    5.5.1 Edge Detection ................................................................................... 66
    5.5.2 Hausdorff Distance Assessment Results for the IRCCyN Database ......... 67
5.5.3 Gradient Based Metric Assessment Results for the IRCCyN Database ....................... 73
5.5.4 Displacement Estimation Assessment ........................................................................... 78
5.5.5 Final Results ................................................................................................................... 78
5.6 Final Evaluation of the Proposed Metrics ............................................................................. 79
5.6.1 Metrics Comparison Using the IRCCyN Database ............................................................ 79
5.6.2 Metrics Comparison Using the SIAT-based Dataset ........................................................... 80

Chapter 6. Summary and Future Directions .................................................................................. 83
6.1 Summary ................................................................................................................................. 83
6.2 Future Work .......................................................................................................................... 84

Appendix A .................................................................................................................................. 85
Bibliography .................................................................................................................................. 89
List of Figures

Figure 1.1 - Promising 3D applications: a) home television; b) video gaming; c) remote surgery; d) surveillance................................................................. 1

Figure 2.1 - 3D Video Transmission Chain .................................................................................................................. 5

Figure 2.2 - Illustration of stereoscopic vision [1], and illustration of parallax: b) view with right eye closed; c) view with left eye closed [2]. .................................................................................. 6

Figure 2.3 - a) Illustration of visual accommodation [3]; b) Illustration of vergence [4] ........................................ 7

Figure 2.4 - Perspective projection geometry [5] ........................................................................................................... 7

Figure 2.5 - Example of parallax resulting from two cameras [6] ................................................................................ 8

Figure 2.6 - A light ray (in red) and its radiance (L) ...................................................................................................... 8

Figure 2.7 - S3D video: example of left and right views [9] .................................................................................................. 9

Figure 2.8 - MVV Representation: example of five views [10] ....................................................................................... 9

Figure 2.9 - Video-plus-depth representation: a) texture image; b) depth map [12] ..................................................... 10

Figure 2.10 - Multiview video-plus-depth representation [13] .................................................................................... 11

Figure 2.11 - LDI representation [14] ......................................................................................................................... 11

Figure 2.12 - Example of point clouds: a) Geo-reference [15]; b) Example of an external surface of an object with point cloud [16]; c) Illustration of light intensity in point clouds [17] ............................................................................. 12

Figure 2.13 - a) Schematic representation of a light field camera; b) A light field image [19] ................................. 12

Figure 2.14 - Example of a GOP structure [20] ............................................................................................................. 13

Figure 2.15 - a) Prediction block done by H.264/AVC; b) Prediction block done by HEVC [21] ............................. 14

Figure 2.16 - Illustration of the different types of frame-compatible formats where the x represents a view and o represents another view [22] ................................................................................ 15

Figure 2.17 - Prediction structure in MPEG-2 Multiview Profile [23] ......................................................................... 15

Figure 2.18 - Temporal/Inter-view prediction structure for MVC [24] ......................................................................... 16

Figure 2.19 - a) General concept of image warping; b) DIBR; c) Hybrid approach using DIBR and image warping [25] .................................................................................................................. 19

Figure 2.20 - Cameras for 3D video capture: a) Stereoscopic acquisition [26]; b) Multiview acquisition with more than two cameras [27]; c) Light field camera (Lytro second generation) [28] .......... 19

Figure 2.21 - Depth camera examples: a) Microsoft Kinect 360 [29]; b) Mesa Imaging’s Swiss Ranger SR4000 [30]; c) Velodyne LiDAR [31] ......................................................................................................... 20

Figure 2.22 - Glasses: a) Red/Cyan glasses [32]; b) polarized lenses glasses [33]; c) LCD shutter glasses [34] ................................................................................................................................. 21

Figure 2.23 - a) Autostereoscopic technology [35]; b) Autostereoscopic LCD [36]; c) Nintendo 3DS [37]; d) Autostereoscopic tablet [38] ........................................................................................................... 21

Figure 2.24 - a) Example of a volumetric display; b) Spinning Mirror example [39] ................................................. 22

Figure 3.1 - Perspective projection [42] ..................................................................................................................... 24

Figure 3.2 - Block diagram of the depth based view synthesis proposed in [40] ..................................................... 24

Figure 3.3 - Basic concept of 3D warping [41] ......................................................................................................... 25
Figure 3.4 – a) Depth map before the post-filtering, where the black stripes represent the blank points and irregular depth changes; b) Depth map after the post-filtering using bilateral filtering [40].

Figure 3.5 – Representation of boundary matting. a) Before matting; b) After matting [40].

Figure 3.6 – Inpainting technique. a) Before inpainting; b) After inpainting [40].

Figure 3.7 – Rendering artifacts: a) Ghosting effect [43]; b) Blur [44]; c) Block effect [45]; d) Geometric distortion [46].

Figure 3.8 – Artifacts resulting from the synthesis of a novel view [47].

Figure 3.9 – Typical location of DIBR specific artifacts [48]: a) Using the DIBR technique described in Section 3.3; b) Using the DIBR technique proposed in [49]; c) Using the DIBR technique proposed in [50].

Figure 3.10 – Block diagram of the proposed view synthesis quality assessment framework [53].

Figure 3.11 – Block diagram of the quality measurement system proposed in [55].

Figure 3.12 – Block diagram of the proposed techniques: a) based on contours distortions analysis; b) based on critical areas analysis [48].

Figure 3.13 – Block diagram of the metric proposed in [56].

Figure 3.14 – Block diagram of the 3DSwIM metric [62].

Figure 3.15 – Block diagram for DSQM metric.

Figure 4.1 – Block Diagram of the Proposed Quality Metric.

Figure 4.2 – Block Diagram of the 2D Image Quality Metric.

Figure 4.3 – Comparing the shift for the same point between: a) Synthesized image; b) Reference image.

Figure 4.4 – Search block illustration.

Figure 4.5 – Motion vectors obtained for each iteration and associated block sizes.

Figure 4.6 – Block diagram of the Hausdorff based approach.

Figure 4.7 – Block Diagram of the shift estimation and compensation.

Figure 4.8 – a) original Image; b) Synthesized image; c) Synthesized image after shift compensation; d) Original image edges; e) Synthesized image edges; f) Synthesized image edges after shift compensation; g) overlap of d) and e); h) Overlap of d) and f).

Figure 4.9 – Distance between sets: a) Edge block extracted from Figure 4.8 g); b) Matrix containing the distances from set A to set B, from the extracted block.

Figure 4.10 – Histogram of h7 and h9.

Figure 4.11 – Histogram of h8.

Figure 4.12 – Block diagram of Gradient based approach.

Figure 4.13 – Illustration of bin disposition to compute the HOG.

Figure 4.14 – Example of image division into blocks: a) Synthesized image; b) Reference image.

Figure 4.15 – Matrices of block-based quality results for: a) 2D image quality metric (Q); b) Structural quality metric (S).

Figure 5.1 – Image synthesis scheme and corresponding source and original images.

Figure 5.2 – An image region extracted from a reference view and from the synthesized views for each DIBR algorithm.
Figure 5.3 – Sequence information and texture/depth QP pairs [51] ........................................... 62
Figure 5.4 - Distribution of MOS scores for the SIAT-based database images. ............................... 63
Figure 5.5 – Logistic function representation.............................................................................. 63
Figure 5.6 – Scatter example and its respective PLCC.................................................................. 64
Figure 5.7 – Canny edge detection: a) Original image; b) Edge image using automatic T; c) Edge image using \( T = 0.2 \); d) Edge image using \( T = 0.15 \); e) Edge image using \( T = 0.1 \); f) Edge image using \( T = 0.05 \). ................................................................. 67
Figure 5.8 – Variations of PLCC of \((h^9, h^3) + \text{SSIM} \) with MOS for different \( \delta \) ...................... 71
Figure 5.9 – Optimized metric values vs MOS scores: a) before the logistic fitting; b) after the fitting. 72
Figure 5.10 – a) Reference image; b) Synthesized image. ............................................................. 73
Figure 5.11 - Objective Score vs MOS for Gradient based approach before and after the logistic fitting: a) and b). .................................................................................................................. 77
Figure 5.12 – Scatter plot of fitted objective scores vs subjective scores referring to the IRCCyN database for the following metrics: a) DSQM; b) 3DSwIM; c) VCIP2013; d) Proposed. ..................... 80
Figure 5.13 - Scatter plot of fitted objective scores vs subjective scores referring to the SIAT-based dataset for the following metrics: a) DSQM; b) 3DSwIM; c) VCIP2013; d) Proposed....................... 82
Figure A.1 – Image extracted from BookArrival sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold................................................................. 85
Figure A.2 – Image extracted from Lovebird1 sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 85
Figure A.3 – Image extracted from Newspaper sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 85
Figure A.4 – Image extracted from Balloons sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 86
Figure A.5 – Image extracted from BookArrival sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 86
Figure A.6 – Image extracted from Dancer sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 86
Figure A.7 – Image extracted from GT Fly sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 87
Figure A.8 – Image extracted from Kendo sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 87
Figure A.9 – Image extracted from Lovebird sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 87
Figure A.10 – Image extracted from Newspaper sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 87
Figure A.11 – Image extracted from Poznan Hall sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 88
Figure A.12 – Image extracted from Poznan Street sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. ................................................................. 88
Figure A.13 – Image extracted from Shark sequence: a) original image; b) Edge image with \( T = 0.1 \); c) Edge image using the adaptive threshold. .................................................................88
List of tables

Table 3.1 – Common artifacts resulting from Depth-Image based View Synthesis ........................................ 28
Table 3.2 – PLCC of the evaluated quality metrics .................................................................................. 34
Table 3.3 – PLCC between DMOS values and objective quality scores .................................................. 35
Table 3.4 – Performance assessment for Case 1. ....................................................................................... 39
Table 3.5 - Performance assessment for Case 2. ...................................................................................... 39
Table 3.6 - PLCC and RMSE between DMOS and objective scores. ...................................................... 41
Table 3.7 – PLCC and RMSE between DMOS and objective scores. ...................................................... 41
Table 3.8 – PLCC, SROCC and RMSE between DMOS and objective scores [66]. ................................. 42
Table 5.1 – IRCCyN images dataset description .................................................................................... 60
Table 5.2 – Considered DIBR techniques ............................................................................................... 61
Table 5.3 – Approaches and parameters for the different modules of the proposed metric. ................. 65
Table 5.4 – Initial parameter values of the displacement estimation method. ........................................ 66
Table 5.5 - PLCC between the MOS scores and Hausdorff distance combined with SSIM. ................. 68
Table 5.6 – PLCC using SSIM as 2D Quality Metric. .............................................................................. 69
Table 5.7 – PLCC using MSE as 2D Quality Metric. ............................................................................... 69
Table 5.8 – PLCC using FSIM as 2D Quality Metric. .............................................................................. 69
Table 5.9 – Impact of α for the three best (h, H) combinations using SSIM as 2D Quality Metric. ....... 70
Table 5.10 - Impact of α for the three best (h, H) combinations using FSIM as 2D Quality Metric. ....... 70
Table 5.11 - Impact of p for the three best (h, H) combinations using the SSIM as 2D metric. .......... 70
Table 5.12 - Impact of p for the three best (h, H) combinations using the FSIM as 2D metric. .......... 71
Table 5.13 – Parameters that maximize PLCC between the metric values and MOS scores. ............... 71
Table 5.14 – Resulting PLCC, Spearman and RMSE for the best combination of metrics and parameters. .............................................................................................................................. 72
Table 5.15 – PLCC using the HOG with the gradient magnitude as vote weight ................................. 74
Table 5.16 - PLCC using the HOG with the number of pixels as vote weight ........................................ 74
Table 5.17 – PLCC for different combinations of 2D Metric and HC technique. .................................... 74
Table 5.18 - Variation of α for the histogram comparison techniques using SSIM as 2D metric. .......... 75
Table 5.19 - Variation of α for the histogram comparison techniques using FSIM as 2D metric. .......... 75
Table 5.20 - PLCC for HOG+SSIM, and varying p. .................................................................................. 76
Table 5.21 - PLCC for HOG+FSIM, and varying p. .................................................................................. 76
Table 5.22 - Best achieved correlation results for the HOG based approach. ........................................ 77
Table 5.23 – Impact on PLCC using different window search sizes ....................................................... 78
Table 5.24 – Impact on PLCC using an hierarchical displacement estimation technique ..................... 78
Table 5.25 – Final Results for the proposed metric. ............................................................................... 78
Table 5.26 – Correlation results between MOS and objective values for conventional 2D Quality Metrics using the IRCCyN database. .................................................................................. 79
Table 5.27 – Correlation results between the subjective and objective quality scores for IRCCyN database. .......................................................................................................................... 79
Table 5.28 – Correlation results between MOS and objective score for traditional 2D Quality Metrics for SIAT-based database

Table 5.29 – Correlation results between the subjective and objective quality scores for SIAT-based database

Table 5.30 – Metrics assessment results for the IRCCyN dataset, using α=0.5 in the proposed solution

Table 5.31 – Metrics assessment results for the SIAT-based dataset, using α=0.5 in the proposed solution
# Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two Dimensional</td>
</tr>
<tr>
<td>2D+Z</td>
<td>2D-Plus-Depth</td>
</tr>
<tr>
<td>3D</td>
<td>Three Dimensional</td>
</tr>
<tr>
<td>3DVC</td>
<td>Three Dimensional Video Coding</td>
</tr>
<tr>
<td>3DSwIM</td>
<td>3D Synthesized View Image Quality Metric</td>
</tr>
<tr>
<td>7D</td>
<td>Seven Dimensional</td>
</tr>
<tr>
<td>AVC</td>
<td>Advanced Video Coding</td>
</tr>
<tr>
<td>ACR</td>
<td>Absolute Categorical Rating</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
</tr>
<tr>
<td>CSF</td>
<td>Contrast Sensitivity Function</td>
</tr>
<tr>
<td>DBIR</td>
<td>Depth Based Image Rendering</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DMOS</td>
<td>Differential Mean Opinion Score</td>
</tr>
<tr>
<td>DVD</td>
<td>Digital Versatile Disc</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete Wavelet Transform</td>
</tr>
<tr>
<td>DSQM</td>
<td>DIBR-Synthesized Image Quality Metric</td>
</tr>
<tr>
<td>ES</td>
<td>Exhaustive Search</td>
</tr>
<tr>
<td>FTV</td>
<td>Free Viewpoint Television</td>
</tr>
<tr>
<td>FVV</td>
<td>Free Viewpoint Video</td>
</tr>
<tr>
<td>GOP</td>
<td>Group of Images</td>
</tr>
<tr>
<td>HEVC</td>
<td>High Efficiency Video Coding</td>
</tr>
<tr>
<td>HSV</td>
<td>Hue, Saturation and Value</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System</td>
</tr>
<tr>
<td>IFC</td>
<td>Information Fidelity Criterion</td>
</tr>
<tr>
<td>ISO/IEC</td>
<td>International Organization for Standardization / International Electrotechnical Commission</td>
</tr>
<tr>
<td>ITU-T</td>
<td>International Telecommunication Union - Telecommunications Standardization Sector</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>KS</td>
<td>Kolmogorov-Smirnov</td>
</tr>
<tr>
<td>LCD</td>
<td>Liquid Crystal Display</td>
</tr>
<tr>
<td>LDI</td>
<td>Layered Depth Image</td>
</tr>
<tr>
<td>LDV</td>
<td>Layered Depth Video</td>
</tr>
<tr>
<td>LIDAR</td>
<td>Light Detection and Ranging</td>
</tr>
<tr>
<td>MOS</td>
<td>Mean Opinion Score</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Pictures Expert Group</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Squared Error</td>
</tr>
<tr>
<td>MSSIM</td>
<td>Multi-Scale Structural Similarity</td>
</tr>
<tr>
<td>MVC</td>
<td>Multiview Video Coding</td>
</tr>
<tr>
<td>MVD</td>
<td>Multiview Video-Plus-Depth</td>
</tr>
<tr>
<td>MV-HEVC</td>
<td>Multiview High Efficiency Video Coding</td>
</tr>
<tr>
<td>MVP</td>
<td>Multiview Profile</td>
</tr>
<tr>
<td>MVV</td>
<td>Multiview Video</td>
</tr>
<tr>
<td>NQM</td>
<td>Noise Quality Measure</td>
</tr>
<tr>
<td>OR</td>
<td>Outlier Ratio</td>
</tr>
<tr>
<td>PC</td>
<td>Paired Comparisons</td>
</tr>
<tr>
<td>PLCC</td>
<td>Person Linear Correlation Coefficient</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assessment</td>
</tr>
<tr>
<td>QA-GoP</td>
<td>Quality Assessment Group of Pictures</td>
</tr>
<tr>
<td>QoE</td>
<td>Quality of Experience</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>S3D</td>
<td>Stereoscopic 3D</td>
</tr>
<tr>
<td>SpCC</td>
<td>Spearman Rank Order Correlation Coefficient</td>
</tr>
<tr>
<td>SSIM</td>
<td>Structural Similarity</td>
</tr>
<tr>
<td>S-T</td>
<td>Spatio-Temporal</td>
</tr>
<tr>
<td>ToF</td>
<td>Time of Flight</td>
</tr>
<tr>
<td>UHD</td>
<td>Ultra-High Definition</td>
</tr>
<tr>
<td>UQI</td>
<td>Universal Quality Index</td>
</tr>
<tr>
<td>Acronym</td>
<td>Full Form</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
</tr>
<tr>
<td>VCEG</td>
<td>Video Coding Experts Group</td>
</tr>
<tr>
<td>V+D</td>
<td>Video-Plus-Depth</td>
</tr>
<tr>
<td>VIF</td>
<td>Visual Information Fidelity</td>
</tr>
<tr>
<td>VQA</td>
<td>Video Quality Assessment</td>
</tr>
<tr>
<td>VQM</td>
<td>Video Quality Metric</td>
</tr>
<tr>
<td>VSQA</td>
<td>View Synthesis Quality Assessment</td>
</tr>
<tr>
<td>VSNR</td>
<td>Visual Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>WSNR</td>
<td>Weighted Signal-to-Noise Ratio</td>
</tr>
</tbody>
</table>
Chapter 1. Introduction

1.1. Context and Motivation

Nowadays, 3D video applications and services are emerging, as a result of several technological advances that have led to a significant reduction in the cost of acquisition devices (e.g., 3D cameras and arrays of cameras) and 3D displays (e.g., stereo TV sets and mobile terminals with 3D enabled capacities). New visualization experiences, such as Free Viewpoint Video (FVV), are now possible providing more immersive, interactive and appealing experiences to the end user than those provided by the classic 2D video. However, applications involving 3D images and video goes beyond entertainment (like films, TV, and games), playing an increasingly important role in other areas, such as medicine, education and surveillance. Figure 1.1 illustrates some examples of current 3D image and video applications. In this context, two promising 3D video representations are the multiview video (MVV) and multiview video plus depth (MVD) formats. The multiview video representation consists in two or more views that are simultaneously acquired from different viewpoints. The transmission of multiview video requires that all views be compressed and transmitted, which is a challenge in terms of memory, computational power and bandwidth. The MVD format consists of multiview video and an associated depth map for each view. This format was proposed to reduce the number of views in comparison with multiview video, thus saving bitrate. In the MVD format, a limited number of views (much lower than in multiview video) of texture and depth videos are encoded, and one or more intermediate views are synthesized for any arbitrary viewpoint. This opens the door for novel applications, such as FVV, where several virtual views are synthesized at the decoder (without requiring any information from the encoder) providing a smooth transition (spatially) from one view to another and a selection by the user of any arbitrary view.

Figure 1.1 - Promising 3D applications: a) home television; b) video gaming; c) remote surgery; d) surveillance.
One of the main barriers to the widespread adoption of MVD video in telecommunications services is the ability to ensure satisfying quality of experience (QoE) to the end user. Still, MVD video is rather demanding in terms of bitrate and thus must be efficiently compressed before being transmitted over the network channel. Furthermore, when the transmission channel has limited bandwidth and time varying characteristics, compression and transmission impairments can lead to artifacts, with a higher subjective (and negative) impact on perceived quality than in conventional 2D video. Also, FVV applications require a huge number of viewpoints (up to a few thousand views) to enable free navigation within the video scene. However, due to bandwidth constraints, only a few views are transmitted and the rest are synthesized in the receiver side. The synthesis of intermediate viewpoints is usually made with a Depth Image Based Rendering (DIBR) technique, which requires the transmission of additional depth data to render intermediate views. Thus, in the synthesized views, several artifacts may occur, due to the lack of accuracy in the acquisition, compression or transmission in the depth maps and errors due to the view synthesis solution (especially relevant in occluded areas). Therefore, the development of accurate and practical methods to assess the MVD video quality is of the upmost importance for content and service providers, who need to optimize the 3D video delivery process. In this process, subjective assessment is expensive, time consuming and it is not possible to conduct in real time, so an objective and automatic reliable quality assessment, which takes into account the human visual system, remains an open problem.

The work developed in this dissertation is related with the MVD video quality assessment field, which is nowadays a very hot topic, since it is necessary to ensure successful MVD systems, that should be constantly monitored and optimized with a reliable quality evaluation procedure. Indeed, the success of 3D video applications, namely FVV, depends on the ability of MVD systems to provide high quality content with a satisfying Quality of Experience (QoE).

1.2. Objectives

The image quality assessment techniques available in the literature have left open doors to explore, regarding the detection of synthesis artifacts and the quantification of their impact on QoE. In this context, the main objective of this dissertation is to propose a new full reference quality assessment metric for DIBR-based synthesized images. This new metric should mainly quantify the distortions introduced in perceptually sensitivity areas, such as edges, and disoccluded regions. To achieve this goal, existing edges distortion metrics - available in the literature and developed with different purposes will be implemented and assessed, for the quality evaluation of synthesized images. The best solution will be combined with an existing 2D metric, for an overall quality metric of synthesized views impaired by synthesis specific artifacts and compression specific artifacts (these last ones inherit from the source views, if compressed).

1.3. Main Contributions

This thesis provides a comparative study of edges distortion metrics - such as the family of Hausdorff distances and the image histogram of oriented gradients (HoG) - to conclude about the best solution for detecting and quantifying DIBR specific artifacts. From this study, a new variant of the Hausdorff
distance is proposed that, when combined with a conventional 2D image quality metric (namely, SSIM), allows a clear improvement in the prediction of the image subjective scores, comparatively to the considered state-of-the-art metrics, developed with the same purposes.

1.4. Thesis Outline

This Thesis is organized in the following chapters:

Chapter 1 explains the context and motivation behind the topic of this dissertation and also the objectives and main contributions.

Chapter 2 reviews the 3D video transmission chain, as well as some basic concepts about 3D video.

Chapter 3 presents an overview of objective quality assessment metrics for MVD systems, with focus in methods to objectively measure the quality of synthesized views. After reviewing the concepts and techniques that can be used to render novel views, a list of the main artifacts generated in this process are presented. Then, six state-of-the-art metrics are described.

Chapter 4 presents a detailed description of the new proposed metric for quality assessment of synthesized images. In this chapter, all the main steps involved on the metric are presented and explained, together with the motivation behind them.

Chapter 5 describes the test material used to evaluate the performance of the proposed metric, as well as the methodology followed in the evaluation. The metric assessment results are also presented and discussed, and compared with state-of-the-art techniques, developed with similar purposes.

Chapter 6 ends the thesis with the final remarks of the presented work and suggestions for future research directions.
Chapter 2. 3D Video Transmission Chain

2.1 Introduction

This Chapter presents an overview of the main building blocks of a 3D video transmission chain (see Figure 2.1), namely:

- **3D Video Capture and Representation**: 3D video is typically acquired by two or more video cameras and depth sensors, but new technologies, as the light field cameras, are emerging; the acquisition solution strongly conditions the video representation. The different 3D video representations can be divided in two main classes: texture-based, if only the color of the real world objects is considered (as in traditional 2D video), and depth-enhanced, if the distance between the acquisition system and the objects is also represented.

- **3D Video Coding/Decoding**: Since 3D video generates a huge amount of data, it has to be efficiently compressed, before being transmitted or stored, using a coding standard adapted to the video representation type; at the receiver, the reverse operation (decoding) is applied to the encoded video.

- **3D Video Rendering**: After decoding, the 3D video needs to be converted into the display format. The texture-based formats can be directly displayed; depth enhanced formats allow to compute virtual views via depth-image-based rendering (DIBR) processing, being more adequate for auto-stereoscopic displays and free-view point video (FVV) systems.

- **3D Video Display**: Several types of 3D displays are available: the classical 2-view stereo with one view for each eye and some kind of glasses (polarization, shutter) to filter the corresponding view, and the auto-stereoscopic displays which do not require glasses; in this case, several views are displayed at the same time and a lenticular sheet or a parallax barrier element in front of the light emitters ensures correct view separation for the viewer’s eyes. Beyond that, development of holographic displays are progressing.

In section 2.2 some basic concepts inherent to 3D vision are introduced; section 2.3 describes the most common 3D video representation and formats, highlighting the associated advantages and drawbacks; the existing standards for 3D video content compression are reviewed in section 2.4; in section 2.5, the concept of 3D video rendering and associated procedures are described; finally, section 2.6 presents the different 3D video capture and displaying technologies.

2.2 Basic Concepts

Humans have the ability to perceive the world in three dimensions (3D) and are able to infer the
relative distance between objects. The main mechanisms of the Human Visual System involved in 3D vision are:

- **Stereoscopic or binocular vision**: consists in acquiring two different views of the same real world scene, each view by each eye, taken from a slightly different angle; depth perception results from the fusion of the two views in the brain. As an example, consider Figure 2.2 (a), where the left and right eyes are focused in two points, A and B, of the 3D world. The optical axes of the two eyes converge to the observed points, forming angles $\phi_a$ and $\phi_b$ (parallactic angles) between them, which are inversely proportional to the distance to the observed point. Automatically, the human brain associates each angle to a distance (in this case, $D_A$ and $D_B$) and the difference between the angles creates the depth perception between the points ($D_B - D_A$).

- **Parallax or Binocular Disparity**: is the apparent displacement of an object when viewed from different locations. As an example, consider Figure 2.2-(b) and (c), where someone is observing the landscape with his finger in front of his eyes, and closing each eye at a time - the apparent motion of the finger results from the parallax or binocular disparity.

- **Visual Accommodation**: this is the process of focusing on an object from different distances by changing the curvature of the human lens (Figure 2.3 (a)). That means that whenever humans look at objects, our eyes are accommodated by an amount that depends on the distance between the eyes and the object of interest. Visual accommodation is an important aspect to have into account for an enjoyable QoE when watching a 3D movie.

- **Vergence**: the movement of eyes when looking to an object. If some object is close to the eyes, the eyes rotate towards each other making a convergent movement, as represented in Figure 2.3 (b); for an object farther away, the eyes rotate away from each other, in a divergent movement. Typically, the accommodation and vergence are intrinsically linked, since the amount of accommodation (to focus an object) depends on the amount of vergence to fixate it.

For a realistic 3D image and video representation of the 3D world, allowing the user to have the perception of depth and the geometry of the objects, 3D cameras and displays should take into account the mechanisms involved in 3D human vision. Nowadays, and despite the remarkable
evolution of 3D equipment, it is not yet possible to perfectly stimulate the HVS to produce a fully realistic and immersive 3D world recreation.

![Image](image1.png)

**Figure 2.3** - a) Illustration of visual accommodation [3]; b) Illustration of vergence [4].

Before going further on 3D video, it is important to understand how cameras acquire the 3D world. An image captured by a camera is a projection of a 3D scene into a 2D plane - as represented in Figure 2.4, when the point Q belonging to the 3D space is captured by a camera, it is projected on point q in the camera’s plane.

![Image](image2.png)

**Figure 2.4** - Perspective projection geometry [5].

The relationship between the 3D coordinates \((X, Y, Z)\) of point Q, and the 2D coordinates \((x, y)\) of its projection, q, is given by:

\[
(x, y) = \frac{f}{Z}(X, Y)
\]

(2.1)

where \(f\) is the camera’s focal length. After 2D projection on the image plane, the depth information of the objects (Z coordinate) is lost; however, if at least two cameras are used, the depth of the 3D objects can be recovered. As an example, consider Figure 2.5, where two parallel cameras acquire points C and D, and parallax (or disparity) is created on the projected images; it can be shown that the depth, Z, of any point captured by both cameras can be found by applying (2.2):

\[
Z = f \frac{b}{p}
\]

(2.2)

where \(b\) is distance between cameras (also known as baseline) and \(p\) is the parallax of the point.
Another important question is how to relate the acquired brightness of a particular point in the image plane, with the light that arrives from the corresponding scene point. Let assume that the fundamental carrier of light is a ray (a valid approximation whenever the objects size is much larger than the light wavelength, $\lambda$); by definition, the measure for the amount of light traveling along a ray is radiance, $L$ ($W\cdot sr^{-1}\cdot m^{-2}$). The plenoptic function, $L(x,y,z,\theta,\phi,\lambda,t)$, firstly proposed in [7], describes the radiance arriving at any point $(x,y,z)$, from any direction $(\theta, \phi)$, at any wavelength $(\lambda)$ and at any time $(t)$ (see Figure 2.6). This plenoptic function represents the light obtained from any 3D spatial position any angular viewing direction, over time and for each wavelength and allows a more complete, realistic and immersive visual experiences. Inspired by these model, new cameras and displays are currently being developed.

During the video acquisition by a 2D camera, the radiance arriving from the scene is focused on a sensor pixel, using focusing optics such as lenses. This focusing is equivalent to integration of the light field along $\theta$, $\phi$, and $t$, as well as along specific wavelength intervals that correspond, typically, to the red (R), green (G) and blue (B) bands of the visible spectrum, giving rise to a 2D color image or video.

### 2.3 3D Video Representation

Nowadays, to provide successful immersive and realistic experiences, efficient 3D video representations are necessary; each representation may have a different impact on the system performance, such as bit rate and quality requirements, end-user QoE and backward compatibility with the 2D video systems [8]. In this section, the most popular 3D video representations are described.
A - Stereoscopic 3D (S3D) Video

Stereoscopy was the first technique deployed to enable depth perception and a three-dimensional effect; it is also the most common 3D video representation, particularly on cinema and television.

Stereoscopic video (S3D video) consists in two views of the same visual scene, acquired with two cameras, positioned in slightly different positions; the resulting images or videos are typically known by left and right views. Each view will be received respectively by each eye and that will result on binocular disparity and, consequently, on perception of depth, as shown in section 2.2. Figure 2.7 is an example of left and right views of a same scene; with a closer look it is possible to see that the left image is shifted to the left, relatively to the right image. The disparity between the left and right images allows the perception of depth in 3D human vision.

![Figure 2.7 - S3D video: example of left and right views](image)

This type of representation has, however, some drawbacks. First, it only adds one depth cue - parallax (or binocular disparity) - and so it does not allow a completely immersive 3D experience; it also requires the use of specialized glasses. Also, the S3D content is normally tuned for specific viewing conditions which can become a significant problem when S3D video produced for cinemas is used in homes, where the viewing conditions can change significantly [8]. Although the generated data is two times the 2D video data, it is possible to exploit the correlation among views (disparity estimation). This format allows direct backward compatibility with 2D displays by using only one of the views.

B - Multiview Video (MVV)

Multiview Video represents the visual scene from different perspectives and is usually acquired with an array of cameras, typically between 2 and 20, normally rearranged in a linear or curvature fashion. Figure 2.8 represents some views that are captured by five cameras, each one with a different perspective of the scene. Naturally, with the increasing number of views the resulting data increases proportionally, so very efficient compression schemes are necessary. In many cases, those views are highly correlated so it is possible to take advantage of the redundancies between each one.

![Figure 2.8 - MVV Representation: example of five views](image)
This type of 3D representation is suited for auto-stereoscopic displays that enable the viewing of two different pairs of images from an arbitrary position in space, thus providing horizontal and (motion) parallax without the need for glasses. Compared to stereoscopic display, MVV allows a better user’s QoE. Usually, and to avoid the acquisition and transmission of many views, it is necessary to perform an accurate synthesis of intermediate views, to drive the auto-stereoscopic displays. Similarly to S3D, MVV also allows backward compatibility with 2D displays by considering only one of the views.

C - Video-Plus-Depth (V+D)

Video-plus-depth, also called 2D-plus-depth or 2D+Z, combines a 2D video (in this context also known as texture channel) with an extra channel corresponding to depth information (see Figure 2.9); this channel, also called depth map, is a grayscale image where each pixel intensity represents the distance between the corresponding 3D point in the scene and the camera image plane. Through suitable algorithms, it is possible to synthesize different views from the texture and depth, using depth image based rendering (DBIR) [11], to create a 3D stereoscopic video. As such, depth perception can be adapted at the receiver end, according to the display characteristics and viewing conditions, minimizing the visual fatigue and improving the user’s QoE. However, dealing efficiently with occlusions and disocclusions remains a challenge. The 2D video component (texture channel) provides backward compatibility with classic 2D video displays; also, the transmission of the extra depth map information only requires a marginal bit rate increase.

Figure 2.9 - Video-plus-depth representation: a) texture image; b) depth map [12].

D - Multiview Video-Plus-Depth (MVD)

The Multiview Video-Plus-Depth (MVD) representation is similar to video-plus-depth but with more than one view; each view is acquired with its associated depth map, as shown in Figure 2.10. This allows a reconstruction of the 3D geometry, and the synthesis of intermediate views (using DBIR), with much better accuracy than the one obtained with MVV or video-plus-depth. In consequence, MVD is suited for auto-stereoscopic displays providing better QoE than Video-Plus-Depth. Like in MVV, the redundancies between views can be exploited; also, texture and depth are jointly encoded which brings higher coding performances. Backward compatibility with 2D displays is also ensured by considering the texture channel of only one view.
E - Layered Depth Video (LDV)

Layered Depth Video (LDV) results from a sequence of Layered Depth Images (LDIs). An LDI is a representation of a 3D scene from a single viewpoint which contains multiple depth pixels (a pixel with associated depth information) at each discrete location of the image. This representation format contains information about occlusion, allowing a fast and more accurate image synthesis of additional viewpoints. Figure 2.11 depicts an LDI representation for a single viewpoint; the first layer is obtained by sampling the first surface seen along the line of sight. Subsequent layers represent the information of the next surfaces along the same line. As also shown in Figure 2.11, an LDI contains information about occluded pixels in the current viewpoint by sampling an image with different layers, so it can be considered as an extension of the image-plus-depth representation [8]. Backward compatibility with 2D displays cannot be obtained.

F - Point Clouds

A point cloud is a collection of 3D samples that represent a real surface with a certain density, where each position is defined by its 3D coordinates (X, Y, Z) and some corresponding attributes. Naturally, the attributes include information about the 3D sample, such as color, normal vectors (vectors perpendicular to the surface), properties of the surface, etc. Depending on the viewpoint, several and different colors can be associated to the same 3D sample, and according to the light that is radiated from the point in the different directions. Point clouds can be acquired using multiple scanners and depth sensors, in several setups, and be used for different purposes (see Figure 2.12), including 3D CAD models, geographic information systems, real-time communications or even medical imaging. Point clouds may have a few thousands or billions of points and can be used to reconstruct and object or a visual scene in a realistic way.
Figure 2.12 - Example of point clouds: (a) Geo-reference [15]; (b) Example of an external surface of an object with point cloud [16]; (c) Illustration of light intensity in point clouds [17].

G - Light Fields

Formally, a light field [18] is defined as the radiance along rays in empty space; it results from the 7D plenoptic function - \( L(x, y, z, \theta, \phi, \lambda, t) \) - defined in section 2.2, by assuming that the radiance along a ray remains constant along its length and along time (so, the \( z \) and \( t \) variables can be discarded), and also that the radiance is measured on the visible part of the electromagnetic spectrum (allowing to discard \( \lambda \)); accordingly, the light field can be considered as a 4D restriction - \( L(x, y, \theta, \phi) \) - of the 7D plenoptic function.

A light field image can be created from an array of cameras, each one capturing the scene from a different perspective; to work efficiently, every camera in the array needs to be perfectly calibrated in terms of focal length, shutter timing, and exposure. Another solution, is to use a light field camera (also called plenoptic camera); this camera has a micro-lens array just in front of the imaging sensor (Figure 2.13 (a)). Such micro-lens array consist of many small lenses, that capture individual light rays, each one corresponding to an observation angle (defined by \( (\theta, \phi) \)). The resulting raw image is a composition of many tiny images (micro-images), as many as micro-lenses (Figure 2.13 (b)). The light field image spatial resolution corresponds to the number of micro-lenses in the light field camera (indexed by \( x, y \)), and the light field angular resolution corresponds to the pixel resolution behind each micro-lens (indexed by \( \theta, \phi \)).

Since each tiny image represents a view from a specific direction, light field images allow to create a large set of 2D images each one corresponding to a different viewpoint, and thereby to obtain a stereoscopic (using two viewpoints) or an auto-stereoscopic (using several viewpoints) representation; 2D backward compatibility is achieved by extracting only one viewpoint and, additionally, users are also able to "refocus" the 2D images at particular distances after the photo was acquired. The main disadvantage of the light field images is the huge amount of data generated, requiring highly efficient compression techniques.

Figure 2.13 – a) Schematic representation of a light field camera; b) A light field image [19].
2.4 3D Video Content Compression

With the increasing number of emerging multimedia services such as 3D Free Viewpoint Television (FTV), 3D stereoscopic and auto-stereoscopic video, including 3D video in mobile devices, it is important to have efficient 3D compression techniques. Several coding solutions that are tailored to the different types of 3D video representation and to the functionalities required by the end-user are nowadays available.

2.4.1 Coding Standards for 2D Video

Most of the 3D video coding standards are extensions of standards developed for 2D video, namely: MPEG-2, H.264/AVC and HEVC; as such, they are reviewed in this section.

- **MPEG-2 Video:** This coding solution was extensively used for 2D video and it is still used in many applications, such as digital television and DVDs (Digital Versatile Discs). This technique exploits inter-frame dependencies to predict the next frames based on the previous and/or the following frames, exploiting temporal correlations; spatial correlation inside a block is exploited through the use of block based DCT (Discrete Cosine Transform) and quantization. The DCT coefficients are quantized to reduce the number of bits required to represent the DCT coefficients. A video sequence is organized into GOPs (Group of Images) with different image types (see Figure 2.14): I (Intra Frames), for which only spatial correlation is exploited; P (Predicted Frames), for which temporal redundancy is exploited, but with the temporal prediction based only on one previous image of type I or P; B (Bidirectional Predicted Frames), for which the temporal prediction is based on one previous and one following image of type I and P (predictions from the two frames are averaged).

![Figure 2.14 - Example of a GOP structure [20].](image)

- **H.264/AVC (MPEG-4):** This coding solution jointly designed by the ITU-T VCEG and ISO/IEC MPEG groups improves MPEG-2 Video, resulting in a similar video quality at half the bit rate. The higher efficiency of H.264/AVC, comparatively to MPEG-2, results from new and improved features, such as multipicture inter-frame prediction, quarter-pixel motion prediction, spatial prediction with novel intra modes (i.e., spatial prediction inside each frame), variable block size, new integer DCT and adaptive entropy coding tools. Nowadays, this coding solution is used in many applications, from digital TV, Blu-ray, videotelephony, etc.

- **HEVC (High Efficiency Video Coding):** This is the most recent coding solution which is also able to double the compression factor for the same quality level with respect to H.264/AVC. HEVC can be used to efficiently compress high-resolution videos, from HD up to 4K/8K UHD. HEVC improves the tools used in previous codecs but the main novelty is the adoption of
larger block structures with more flexible mechanisms of subdivision. In Figure 2.15 it is shown a comparison between the prediction block sizes in H.264/AVC and HEVC and it is easily concluded that in HEVC blocks can have a larger size and can be decomposed more times. However, in HEVC, more tools were improved, namely: intra prediction with more angular modes, improved motion compensated filters, motion vector prediction, sample adaptive offset filtering to improve the quality of the prediction, etc. A drawback of this technology is that it increases the complexity significantly with and thus higher processing power is needed to encode and decode HEVC streams.

![Figure 2.15](image)

Figure 2.15 - a) Prediction block done by H.264/AVC; b) Prediction block done by HEVC [21].

### 2.4.2 Simulcast Coding for 3D Video

Simulcast is the oldest and simplest coding technique that employs classical single-view video codecs to compress stereoscopic or multiview video content. This solution encodes and decodes each view independently, using available codecs for 2D video (such as MPEG-2, H.264/AVC or HEVC) and thus, it does not exploits the inter-view correlation between images taken from different perspectives. However, to achieve improved performance and better bit rate savings, it is possible to account for the *binocular suppression theory*; according to this theory, if two views have different quality then the perceived quality is closest to the higher quality view, meaning that one of the views can be encoded in a lower resolution (and quality). The main advantage of this technique is its simplicity due to the fact that each view is coded independently of each other. Another advantage is the backward compatibility with the 2D displays, for which only one view should be considered.

### 2.4.3 Frame-compatible Stereo Interleaving

Frame-compatible stereo interleaving consists in multiplexing the left and the right views into a single frame, e.g. through temporal or spatial interleaving. There are a few options for the spatial multiplexing, as shown in Figure 2.16, where the circles and crosses represent left or right view. The top-bottom and side-by-side are the most commonly used formats as they produce a better visual quality compared with row interleaved, column interleaved and checkboard [8]. Another format is the temporal multiplexing which interleave left and right views by alternating the frames, in this case, for stereo representations the number of displayed frames per second is half than spatial interleaving loosing spatial resolution but with the advantage of preserving the full resolution of each image [8].
The major benefit of stereo interleaving is that it represents stereoscopic video in a way that it is compatible with the existing 2D codecs. Naturally due to subsampling (to obtain the left and right views in a lower resolution) the video quality is reduced and it is also necessary to signal the subsampling so that the devices can decode and distinguish the left and the right views which is essential to correctly display the stereoscopic content.

Figure 2.16 – Illustration of the different types of frame-compatible formats where the x represents a view and o represents another view [22].

2.4.4 Texture Based Coding Standards for 3D Video

Texture-based coding standards are codecs that exploit the correlation between views in addition to the temporal and spatial correlation inside each view and are typically used for stereoscopic and multiview video coding. The main texture-based coding standards are the MPEG-2 Multiview Profile (MVP), the Multiview Video Coding (MVC) and the Multiview High Efficiency Video Coding (MV-HEVC) which are described below.

- **MPEG-2 Multiview Profile (MVP):** This profile extends the MPEG-2 video coding solution to code stereoscopic video. One of the main features is the use of scalable coding tools, which consists in the capacity of extracting from a single bitstream, different layers, which correspond to increasing video qualities and resolutions, but still guaranteeing compatibility with the MPEG-2 main profile with a base layer. In MVP, one of the views is coded in a backward compatible base layer and, for the other view, an enhancement layer can be used. In Figure 2.17 it can be observed that the base layer (upper view) is coded with a standard MPEG-2 non scalable codec, to maintain compatibility with 2D devices, and the enhancement layer exploits inter-view correlation between enhancement and base layers.

Figure 2.17 - Prediction structure in MPEG-2 Multiview Profile [23].
- **Multiview Video Coding (MVC):** This format is an extension of H.264/AVC and was created to code stereoscopic and multiview 3D video. Naturally, when multiview is used there are a lot of redundancies between views which can be exploited to improve the coding efficiency. Figure 2.18 illustrates the prediction structure for MVC, where inter-view statistical dependencies are exploited combining temporal and inter-view prediction. The base view (View 0 of Figure 2.18) only exploits temporal redundancies in order to maintain the compatibility with 2D decoders, and the other views exploit not only temporal but also inter-view prediction.

![Figure 2.18 - Temporal/inter-view prediction structure for MVC [24].](image)

- **MV-HEVC (Multiview High Efficiency Video Coding):** HEVC has also defined a multiview extension in a similar way to MVC. This representation reuses the coding tools of HEVC and permits efficient coding of multiple views. The novelty and the major advantage of this codec is the use of reference frames of other views to perform motion-compensated prediction. This is a major advantage compared with HEVC simulcast that allows to achieve higher bit rate savings.

### 2.4.5 Texture-plus-depth Based Coding Standards for 3D Video

Although the 3D video coding standards already described achieve a good compression rate, the bit rate still grows linearly with the number of encoded views [13]. Another alternative is to add a depth layer for each view, and to use depth-image based rendering (DIBR) to synthesize intermediate views based on the existing (and transmitted) views and their respective depth maps. Thus, depth based representations allow a more efficient coding than pure texture based representations since not all of the views need to be coded and transmitted. The following two texture-plus-depth based video coding standards have been developed:

- **MPEG-C Part 3:** This coding standard allows to encode the video plus-depth format but also provides backward compatibility with 2D devices that support MPEG-2 or H.264/AVC. This technique does not introduce any specific coding algorithm for depth or texture - it allows the encoding of depth map sequences with available (and even future) 2D video compression standards. Thus, only a high-level syntax is specified to allow the decoder to correctly distinguish texture and depth data. A depth map can be interpreted as a grey scale color image to describe the distance of each object to the camera. In this coding solution a depth map is compressed conventionally like a texture image. Due to the depth map characteristics,
depth maps require less rate than a texture image. Some coding experiences revealed that the bit rate need for depth is 10 to 20% of the texture bit rate [8].

- **3D-HEVC:** This extension of the HEVC standard or, more precisely, an extension from MV-HEVC, provides increased coding efficiency compared with MV-HEVC by encoding depth and texture. In this codec, the views are divided in independent views and dependent views. The independent views are coded using the tools of HEVC codec; for the dependent views, some tools were added to achieve higher coding efficiency. The coding rate savings of 3D-HEVC relatively to MV-HEVC vary from 10 to 40%; this gain is mainly justified by the novel tools for exploiting the correlation between depth and texture layers, besides the tools already existing on the HEVC standard.

### 2.5 3D Video Rendering

In the context of this report, 3D video rendering consists on the creation of novel views of the same visual scene, from previously acquired and transmitted views, with slightly different viewpoints. The main goal is to optimize the user QoE for 3D video applications that require multiple views, while minimizing the number of transmitted views and, consequently, the required transmission bandwidth.

#### 2.5.1 Common Applications

3D video rendering is typically used for representations based on texture plus depth (e.g., V+D and MVD), but can also be used for pure texture based representations (e.g., MVV); it is a key procedure in the following applications:

- **Stereoscopic video:** when the V+D representation is used, and since it only provides one texture image and its correspondent depth map, it is necessary to render the texture image for the additional view (left or right, depending on the transmitted view).

- **Autostereoscopic video:** when MVD or MVV are used with autostereoscopic displays, to provide a full 3D experience independent of the user’s viewing position, a dense set of views of the same visual scene are required. However, the transmission bandwidth costs increase with the number of transmitted views, and therefore, also the cost of the system. To limit the number of transmitted views, it is necessary to synthesize new images of different viewpoints at the receiving side, using the adjacent views.

- **Free Viewpoint Video (FVV):** this application allows the user to choose the viewpoint of a visual scene. In this case, image rendering is used to generate new views from any 3D position, based on a set of views acquired (and transmitted) by a limited number of cameras placed around the scene.

#### 2.5.2 Common Approaches for 3D Video Rendering

Several solutions for image and video rendering have been proposed in the literature; this sub-section presents the main approaches that are typically associated with the applications listed above:
- **Image warping**: in this technique (also known as image morphing) synthesis is done in the 2D image space (see Figure 2.19 (a)). When rendering a virtual view $V$ between images $A$ and $B$, it is assumed that the correspondence map between $A$ and $B$ is known; this map stores for each pixel position, $x_i$, in $A$, the vector pointing to the corresponding location, $x_j$, in $B$; this correspondence information can then be used to project each pixel from $A$ and $B$, toward its corresponding pixel in $V$. $V$ is then obtained as a weighted combination of $A$ and $B$ (in Figure 2.19 (a), the parameter $b$ represents the used weight) [25]. The main drawback of this technique is the hole problem due to occlusions and dis-occlusions of pixels when synthesizing the new viewpoint, which may result in an unpleasant visual effect. Furthermore, correspondence estimation in 2D is a weak constrained problem.

- **Depth Image Based Rendering (DIBR)**: in this technique (also known as 3D warping), the view synthesis is based on the color and depth information of the available view(s). Figure 2.19 (b) illustrates the concept: for any given pixel location, $x_i$, in image $A$, its original 3D location, $X_i$, is computed using the depth information and the camera parameters; to synthesize the view of a virtual image, each pixel of an image $A$ is re-projected to the image plane of $V$, using the camera parameters of $V$. In Figure 2.19 (b), $C_I(.)$ denotes the projection from 3D world space to the image space of $I$ and $C_I^{-1}(.)$ is the inverse projection. For a better synthesis, the rendered image may be obtained from a combination of two or more re-projected views ($A$, $B$ in Figure 2.19 (b)). The main drawbacks of this technique are the need for an accurate camera calibration procedure (whose parameters must be known) and the production of holes and ghost artifacts in the generated views. Some techniques (e.g. inpainting) may be used to reduce this unpleasant artifact, at the expense of additional complexity and processing time.

- **Hybrid (image warping + DIBR)**: This method is a mix of image warping and DIBR and it seeks to reduce the image artifacts due to the occlusions; the key idea is to apply image warping in the 3D domain. Similar to warping-based rendering, intermediate views are rendered by interpolating pixel positions, but those positions are now defined on the 3D world. Corresponding pixels positions, $x_i$, $x_j$, of the known views ($A$ and $B$ of Figure 2.19 (c)) are first projected in their world space positions, $X_i$, $X_j$; the final pixel location (on the synthesized view, $V$) is obtained by re-projecting the interpolated position to the image plane of $V$. The main drawback of this technique is the long processing time, invalidating the approach for real-time applications [25].
2.6 3D Video Capture and Display Technologies

This section presents an overview of the technologies to acquire and display the different forms of 3D video representation.

2.6.1 3D Video Capture

The 3D video capture systems can be divided in two main classes, according to the video channel - texture and depth - that is being acquired.

A - Texture Acquisition

The oldest method in 3D video capture is to use two cameras, or a camera with two lens (Figure 2.20 (a)), producing a pair of 2D videos - known as stereoscopic video - each one with a slight different perspective of the 3D scene. However, the use of more than two cameras (Figure 2.20 (b)), resulting in the so called multiview video, is becoming quite popular. Typically, multiview video is captured using up to 20 cameras in a linear or non-linear arrangement, with each camera viewing the scene from a different angle.

A more recent solution for 3D video capture is a light field (microlens) camera. A light field camera has a physical array of micro lens into the optical path of a standard 2D sensor, to capture different directions of light; the Lytro consumer cameras of Figure 2.20 (c) follow this model. Also, dense 2D arrays of cameras (up to 200 cameras with short baseline distances) can be used to obtain one of the 3D video representation models, in this case providing full parallax.

B - Depth Acquisition

To acquire the depth of a visual scene an infrared cameras can be used. These cameras allow to...
compute the distance from the camera to the objects when a structured light projector is used to illuminate the visual scene. A well-known example of this technology is the Kinect camera, from Microsoft (Figure 2.21 (a)). The scene depth can also be acquired with the Time of Flight (ToF) imaging cameras, which measure the propagation time between the camera and each point of the scene of a continuous light signal, resulting in a depth map (Figure 2.21 (b)). Another technology is a LIDAR sensor (Figure 2.21 (c)), which measures the distance to the objects by emitting laser pulses and analyzing the reflected signal. This technology is also used in geographical information system applications, or in autonomous vehicles, to measure the distance between the car and the obstacles in the road.

Figure 2.21 - Depth camera examples: a) Microsoft Kinect 360 [29]; b) Mesa Imaging’s Swiss Ranger SR4000 [30]; c) Velodyne LIDAR [31].

2.6.2 3D Video Displays

The main requirement for a 3D display is to provide a pleasant immersive 3D experience, with a real depth perception; three main technologies are available: stereoscopic displays, autostereoscopic displays and volumetric displays.

A – Stereoscopic Displays

Nowadays, stereoscopic displays are the most popular and low cost solution to enable 3D perception. With this technology it is necessary for the viewer to use specialized glasses that allow the left and right views to be seen by the left and right eyes, respectively. Accordingly, although both views (left and right) are projected in the screen, the glasses demultiplex each view to each corresponding eye. There are different types of glasses:

- **Red/cyan (anaglyph) glasses**: these glasses (Figure 2.22 (a)), use filters of different (usually chromatically opposite) colors, typically red or cyan, and the view intended for each eye corresponds to a filtered colored image with the corresponding lens color; thus, each eye can only see one of the views.

- **Polarized lenses glasses**: the left and right views are projected superimposed on the same display but polarized in orthogonal propagation directions. Since each filter only passes light which is polarized in the same way, each eye only sees the corresponding view and blocks the other one; these glasses are shown in Figure 2.22 (b).

- **Liquid Cristal Display (LCD) shutter glasses**: this solution consists on the use of synchronized shutters associated to the lens, so that when one of them is open, allowing the light to propagate, the other blocks the light. For this solution to work properly, it is necessary to have the glasses and the screen display well synchronized, alternating the views for the
right and left eye at a rate of at least 120 Hz. Figure 2.22 (c) presents a LCD shutter glasses.

**Figure 2.22** – Glasses: a) Red/Cyan glasses [32]; b) polarized lenses glasses [33]; c) LCD shutter glasses [34].

### B – Autostereoscopic Displays

The possibility of providing a real depth perception to the user, without the need of extra equipment, like glasses, may provide a new experience to the users and bring a wider acceptance of 3D video technology. Autostereoscopic displays attempt to achieve binocular vision using screens which can direct the left/right view to the left/right eye, respectively, without the need of special glasses. Autostereoscopic displays use transparent lenticular lenses which appropriately refract the image in such a way that, depending on the view angle, only the correspondent view is visible by each eye. Figure 2.23 (a) shows the autostereoscopic technology using lenticular lens, where the red and blue strips represent, respectively, the left and the right views. Autostereoscopic displays for home television, like LCD or plasma television (see (Figure 2.23 (b))), still have a considerably high cost when compared with the standard 2D home televisions so there has not been a strong adherence yet. Some video consoles, like Nintendo 3DS (see Figure 2.23 (c)), use an autostereoscopic display to enable the depth perception without using specialized glasses; this is probably the most popular autostereoscopic display since it provides a good QoE to the user, for the gaming purpose, at a reasonable price. There are also mobile phones and tablets that have autostereoscopic displays (see Figure 2.23 (d)). Although some of them can reach a satisfactory QoE, the majority were discontinued principally due to the high battery consumption. In conclusion, autostereoscopic devices do not have the user’s acceptance due to several limitations that make users prefer the standard 2D devices, such as: price, bad visual effects which cause fatigue and lack of interesting 3D content.

**Figure 2.23** – a) Autostereoscopic technology [35]; b) Autostereoscopic LCD [36]; c) Nintendo 3DS [37]; d) Autostereoscopic tablet [38].

### C – Volumetric Displays

Volumetric Displays produce an image by projecting light inside a physical volume, using the principles of light diffusion, i.e. emission, scattering, or relaying of illumination to well defined regions. These displays reproduce a visual representation of semitransparent full 3D objects which the user can examine from different angles, like a 3D object, thus providing full parallax. This display still has some limitations, such as: the large amount of bandwidth required to feed imagery to a volumetric display, the inability of reconstructing scenes with viewer-position-dependent effects, such as occlusion and opacity and difficulty in reproducing colors in a realistic way. Figure 2.24 presents an
example of a volumetric display. The most famous display of this family is the holographic displays which may use powerful lasers and plasma materials to produce images that appear from thin air. Other displays work by projecting video with a high frame rate onto a rapidly spinning mirror. As the mirror turns, a different and accurate perspective is projected to each potential viewer in a certain position of space.

![Example of a volumetric display](image1)

![Spinning Mirror example](image2)

*Figure 2.24 – a) Example of a volumetric display; b) Spinning Mirror example [39].*
Chapter 3. Objective Quality Assessment of 3D Synthesized Videos

3.1 Introduction

Three dimensional (3D) video provides a more realistic impression of the scene to the user and enables a new set of advanced functionalities, such as Free Viewpoint Video (FVV). In this case, to reduce the number of transmitted views, and the required transmission bandwidth, intermediate views should be rendered (i.e., synthesized) from the decoded views, allowing a realistic free viewpoint navigation without discontinuities in the scene. To guarantee an adequate QoE to the end-user, the quality of the synthesized views should be evaluated; this will allow: i) the encoder to decide about which views should be transmitted; ii) the decoder side to ask for additional views, if necessary; iii) the service provider to track the media quality that is being delivered to the end-user. Essential to those procedures is the availability of an objective (i.e., automatic) video quality assessment metric, providing quality scores well correlated with the ones resulting from subjective assessment (i.e., using human evaluators).

The main objective of this chapter is to present an overview of the objective quality assessment metrics for 3D synthesized videos. In section 3.2, the perspective projection equations that relate the 3D world coordinates of an object with the resulting 2D coordinates in the image plane, are presented. In order to get a better insight about the artifacts (e.g., their origin and characteristics) resulting from the rendering process, the DBIR technique proposed in [40] and adopted by MPEG as a reference image rendering technique, is described in section 3.3. Section 3.4 presents the principal artifacts resultant from DIBR technique. Finally, section 3.5 reviews the most common 2D quality assessment metrics, and some of the recently proposed quality assessment techniques for 3D synthesized views.

3.2 Perspective Projection

In section 2.2, the perspective projection geometry was introduced assuming the simple case of a camera whose coordinate system coincides with the world coordinate system; this section considers the more general case in which the world and camera coordinate systems are different. Let \( M = (X, Y, Z) \) be the coordinates of point \( M \) in the world reference system and \( m = [u, v] \) the coordinates of its projection, \( m \), in the camera image plane (see Figure 3.1); let also \( \tilde{M} = [X, Y, Z, 1]^T \) be the homogeneous coordinates of point \( M \), in the world reference system and \( \tilde{m} = [u, v, 1]^T \) its homogeneous coordinates in the image plane. Under perspective projection and modeling the camera as the usual pinhole, the relationship between \( \tilde{M} \) and \( \tilde{m} \) is given by the perspective transformation [41]:

\[
\lambda \tilde{m} = \tilde{P} \tilde{M}
\]

where \( \lambda \) is a non-zero scale factor and \( \tilde{P} \) is the camera matrix, given by (3.2):

\[
\tilde{P} = A[R|t]
\]
Matrix $A$ depends only on the camera intrinsic parameters, and has the form (3.3):

$$A = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where:

- $f_u = -f k_u$, $f_v = -f k_v$ are the focal lengths (in pixels) along the $u$ and $v$ axes, respectively;
- $f$ is the focal length in millimeters;
- $k_u$, $k_v$ are the number of pixels per millimeter along the $u$ and $v$ axes, respectively;
- $u_0$, $v_0$ are the coordinates of the image center.

The camera position and orientation (extrinsic parameters) relatively to the world system are represented by the translation vector $t$ and by the 3x3 rotation matrix $R$, respectively.

![Perspective projection](image)

**Figure 3.1 – Perspective projection [42].**

### 3.3 View Synthesis Through Depth-Image Based Rendering

As seen in section 2.5, depth-based image rendering (DIBR) consists in the synthesis of a virtual view from its nearest transmitted views, using their texture and depth channels. In [40], the authors proposed a DIBR algorithm to be used in the context of the FVV applications, which has become the starting point for a MPEG reference view rendering technique. This algorithm combines a set of procedures that are schematically presented in the block diagram of Figure 3.2.

![Block diagram](image)

**Figure 3.2 – Block diagram of the depth based view synthesis proposed in [40].**

The first two modules are described in section 3.3.1 and 3.3.2 while the last two modules are described in section 3.3.3.
3.3.1 Depth Map Projection with 3D Warping

In 3D warping, depth map values in the reference view are back-projected to the 3D world and then re-projected on the virtual image plane (see Figure 3.3); by applying (3.2) and after simple algebra manipulations, those transforms can be described as:

- **Back-projection:**
  \[ [X, Y, Z]^T = R_{\text{ref}}^{-1}A_{\text{ref}}^{-1}[u, v, 1]^T d_{u,v} - t_{\text{ref}} \] (3.4)

- **Re-projection:**
  \[ [U, V, 1]^T d_{U,V} = A_{\text{vir}} R_{\text{vir}} ([X, Y, Z]^T + t_{\text{vir}}) \] (3.5)

where:
- \( R_{\text{ref}}, t_{\text{ref}}, A_{\text{ref}} \) are the camera matrices of the reference view;
- \( R_{\text{vir}}, t_{\text{vir}}, A_{\text{vir}} \) are the camera matrices of the virtual view;
- \( d_{u,v} \) is the depth value at pixel \((u, v)\) of the reference view;
- \( U, V \) are the image coordinates of the virtual view;
- \( d_{U,V} \) is the depth value (re-projected) at position \((U, V)\) of the reference view.

If more than one depth value is re-projected to the same position of the virtual view, the smallest depth value will be kept; only the two nearest views of the virtual one are used, corresponding to the nearest left and right views. The output of this module is equivalent to the depth acquired in the synthesized view.

3.3.2 Post-filtering on the Projected Depth Maps

In most cases, the rendered image shows some artifacts, resulting in an un-natural looking; these artifacts can be reduced by applying a suitable smooth filter. However, instead of smooth/post-filtering the rendered image (i.e., the virtual view texture), which may give rise to a blurred effect; the authors of [40] propose to apply the post-filtering to the depth maps. The principal artifacts to take into account on the projected depth maps are:

- **Blank points:** since the \((U,V)\) coordinates are calculated decimally and rounded-off to the nearest integer value, sometimes it can happen that for some point there are no
correspondent coordinates, which results on a blank point; to solve this problem, median filter is proposed. Blank regions may also appear on the depth map due to disocclusions; in this case, the error can be solved using the projected depth from the other reference view.

- **Irregular depth changes:** this artifact can be reduced using a low-pass filter; however, traditional low-pass filter may result on the edges fading. To preserve the edges, a solution relying on bilateral low-pass filtering should be used [40].

Figure 3.4 shows the difference on the depth map before and after the post-filtering, which results on a smoother depth image.

![Figure 3.4](image)

**Figure 3.4** – a) Depth map before the post-filtering, where the black stripes represent the blank points and irregular depth changes; b) Depth map after the post-filtering using bilateral filtering [40].

### 3.3.3 Rendering Using Boundary Matting and Inpainting

After doing the post-filtering on depth maps, these maps are used to project the positions of each pixel \((U, V)\) of the virtual image plane to each reference camera plane, by applying an inverse 3D warping (module name is referring real camera images in Figure 3.1). Thus, two corresponding pixels, \((u_L, v_L)\) and \((u_r, v_r)\), associated to the left and right reference views are obtained. In order to deal with the occlusions, an occlusion map is obtained by labeling, according to eq. (3.6), the area where the projected depth value, after post-filtering, is less than a pre-defined threshold. Left-side occlusion is assumed to happen when the depth computed from the right side view is small (closest to the camera) and vice versa. The area where \(occ(U, V) = 1\) is considered as the occluded area.

\[
occ_L(U, V) = \begin{cases} 
1, & Z_r(U, V) < \text{threshold} \\
0, & Z_r(U, V) > \text{threshold}
\end{cases}
\]

\[
occ_R(U, V) = \begin{cases} 
1, & Z_L(U, V) < \text{threshold} \\
0, & Z_L(U, V) > \text{threshold}
\end{cases}
\]  

(3.6)

In eq. (3.6):
- \(occ_L\) and \(occ_R\) are the occlusion maps, which are constructed for the left and right reference images;
- \(Z_L\) and \(Z_r\) are the depth values projected from the left and right reference images.

In order to compute the pixel value of the virtual image at position \((U, V)\), eq. (3.7) and (3.8) are used:
\[
I(U, V) = \begin{cases}
(1 - \alpha)I_L(u_L, v_L) + \alpha I_R(u_R, v_R), & \text{occ}_L(U, V) = 0, \text{occ}_R(U, V) = 0 \\
I_L(u_L, v_L), & \text{occ}_L(U, V) = 0, \text{occ}_R(U, V) = 1 \\
I_R(u_R, v_R), & \text{occ}_L(U, V) = 1, \text{occ}_R(U, V) = 0 \\
0, & \text{occ}_L(U, V) = 1, \text{occ}_R(U, V) = 1
\end{cases}
\]  

(3.7)

\[
\alpha = \frac{|t - t_L|}{|t - t_L| + |t - t_R|}
\]  

(3.8)

where

- \( I(U, V) \) means the pixel value at pixel \((U, V)\) of the virtual view;
- \( \alpha \) is a weighting coefficient;
- \( t \) is the translation vector of the extrinsic matrix of the virtual camera, \( t_L \) and \( t_R \) correspond to the translation vector of left and right reference cameras.

This approach assumes that if there are no occlusions, the pixel value should be a weighted average of the corresponding pixels from the left and right reference images; however, if only one of the pixels is occluded, the other non-occluded pixel should be used. Finally, if both pixels are occluded, it will be not possible to make a correspondence so the pixel is left in blank. However, the objects on the rendered view may be barely defined at the borders giving a sensation of shadow around them (see Figure 3.5 (a)); to reduce this problem, the use of boundary matting is suggested, which consists on expanding the border of the occlusion so that the shadow is erased (see Figure 3.5 (b)). After matting, the remaining blank points are filled using an inpainting technique [40]. This technique consists on prolonging the texture from its boundary in order to fill the existing gap. Figure 3.6 shows an application of the inpainting technique.

![Figure 3.5](image1.png)  
**Figure 3.5** – Representation of boundary matting. (a) Before matting; (b) After matting [40].

![Figure 3.6](image2.png)  
**Figure 3.6** – Inpainting technique. (a) Before inpainting; (b) After inpainting [40].
3.4 Artifacts Resulting from Depth-Image Based View Synthesis

The virtual views resulting from the depth-image based synthesis process typically contain different types of artifacts that may impact the users’ quality of experience (QoE); Table 3.1 summarizes the most common artifacts, with their characterization and origin; Figure 3.7 illustrates some of them.

<table>
<thead>
<tr>
<th>Type</th>
<th>Characterization</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ghosting effect</td>
<td>A shadow-like artifact around contours</td>
<td>Not effective inpainting of disoccluded areas</td>
</tr>
<tr>
<td>Region shift /resize</td>
<td>Change of contours position</td>
<td>Not effective inpainting Depth map errors (due to estimation and compression)</td>
</tr>
<tr>
<td>Geometric distortion</td>
<td>Distortion around object boundaries resulting in an unnatural aspect</td>
<td>Not effective inpainting Depth map errors</td>
</tr>
<tr>
<td>Blur</td>
<td>Loss of sharpness resulting on an unclear and cloudy image region</td>
<td>Not effective inpainting Texture compression</td>
</tr>
<tr>
<td>Block effect</td>
<td>Artificial discontinuities with a squared aspect</td>
<td>Texture compression</td>
</tr>
<tr>
<td>Temporal Flickering</td>
<td>Pixel values vary along motion trajectory</td>
<td>Lossy compression Depth map errors</td>
</tr>
</tbody>
</table>

Figure 3.7 – Rendering artifacts: a) Ghosting effect [43]; b) Blur [44]; c) Block effect [45]; d) Geometric distortion [46].

The ghost effect, which is one of the most frequent in the synthesized views, can be classified as fat ghost, skinny ghost and standing-out hole edge (see Figure 3.8). The first two are caused by inadequate hole filling process; standing-out hole edges are caused by color differences between left and right views.

![Figure 3.8 – Artifacts resulting from the synthesis of a novel view [47].](image)

Another important aspect is the artifacts location, since this information, if a priori known, can be explored when developing an objective quality assessment model for synthesized views. Figure 3.9
shows the artifacts location for three different DIBR techniques, allowing to conclude that they are often located along object borders, where disocclusions are more likely to occur.

![Figure 3.9 – Typical location of DIBR specific artifacts [48]: a) Using the DIBR technique described in section 3.3; b) Using the DIBR technique proposed in [49]; c) Using the DIBR technique proposed in [50].](image)

### 3.5 Quality Assessment Models

For the full acceptance of MVD based video applications, such as FVV and autostereoscopic video, high-quality visual content and immersive experience should be guaranteed; this requires the development of objective video quality metrics for MVD services, which may allow to evaluate different parts of a complete system or be employed to optimize already available systems:

- **Performance evaluation of compression algorithms** – The performance of an MVD compression algorithm is typically evaluated by measuring the quality of the synthesized views using original reference views, i.e. an intermediate view is synthesized using the original texture and depth maps, and its distorted version is synthesized from the decoded texture and depth maps (thus with some artifacts); the reference view and the distorted view are then compared using an objective quality metric.

- **Performance evaluation of a DIBR technique** – As seen before, any DIBR technique introduces artifacts on the synthesized views; the performance of the DIBR technique can be evaluated by comparing the quality of the views synthesized using the original texture and depth maps, with the quality of the corresponding original views (if they are available).

- **Selection of the views to be transmitted** – In MVD, the selection of the transmitted views is achieved by a trade-off between video quality and required bitrate - the quality prediction of the intermediated views, that have to be synthesized at the decoder side from the decoded views, requires the existence of an accurate objective metric. This quality assessment can also be used in the process of the multiview acquisition, to decide the most convenient placement of the cameras, and their number.

- **Decoder request for additional views** – At the decoder side, if the quality of the synthesized videos is considered not sufficient to provide an adequate level of QoE, additional views could be requested to the encoder side; once more, an accurate objective quality metric needs to be available.

- **End-to-end quality control** – The network operators and the service providers should be
able to assess, to predict and possibly to control, the end-to-end perceptual quality of the delivered service, and to allocate the network resources according to the required QoE.

Nowadays, there are many objective video quality assessment (VQA) metrics developed for 2D video. However, they are not accurate enough to assess the visual quality of the 3D synthesized views. The 2D VQA metrics underestimate some dominant distortions of the synthesized view such as flickering and inconsistent object shifting and edge distortion, which are very annoying and noticeable for the end user. Also, tiny geometric distortions, consistent object shifting, inter-view camera noise and illumination difference, which can hardly be perceived by human subjects, might be overestimated by traditional VQA metrics [48] [51]. In consequence, several VQA models for 3D synthesized views have been recently proposed (e.g., [48] [51] [52] [53] [54] [55] [56]), and some key findings have emerged that, besides justifying the rational for the proposed methods, may also drive the development of new quality metrics. These findings are summarized below.

A. Applicability of 2D video quality assessment to DIBR synthesized views:
   1. Subjective assessment procedures developed for 2D video are still valid for the subjective quality assessment of DIBR synthesized views; Paired Comparisons (PC) and Absolute Categorical Rating (ACR) results are highly correlated, but fewer observers are required for PC [48].
   2. Objective assessment metrics developed for 2D video hardly correlate with subjective assessments of DIBR synthesized views. This is because artifacts resulting from DIBR algorithms are different from those resulting from compression, and are located in specific areas (those resulting from compression are typically scattered) [48].

B. DIBR artifacts origin:
   1. Artifacts are mainly due to disocclusions which are more likely to occur around sharp discontinuities (objects borders); disoccluded areas have to be extrapolated, through inpainting techniques, resulting in contours discontinuities and texture blurring [48].
   2. Incorrect depth values (due to inaccurate depth estimation or/and depth compression) lead to objects shifts and/or resizing in the synthesized views; however, "consistent" shifts/resize (i.e., the object maintains its original structure), have little impact on the perceived quality [52], [56].
   3. View synthesis yields high distortions that mask those due to depth map compression, suggesting that depth map can be compressed at low bit rates [54].
   4. Texture compression results in blur and block effects on the reference views, that are inherited by the synthesized views [51].
   5. Lack of temporal consistency between successive frames of the synthesized views creates flickering [53].

C. DIBR artifacts versus cameras position:
   1. With the increasing distance between two consecutive cameras the quality of the
intermediate synthesized views decreases [54].

2. Low depth areas are more vulnerable to rendering distortions than high depth areas [53].

D. Perception of DIBR specific artifacts by the Human Visual System (HVS):

1. The main distortions affecting the perceived quality are ghost-type errors, inconsistent object shifts and temporal flickering [51] [56].

2. The HVS is more affected by distortions happening on the front part of the scene (lower depth) [55].

3. The HVS sensitivity in flat areas is higher than in textured regions (areas with high spatial activity); however, errors in object contours have high perceptual impact [51] [55].

4. The synthesis distortions around the object contours are more noticeable for moving objects than for static ones [51].

5. Viewers are very sensitive to artifacts occurring around human representations; a higher weight should be applied to errors in human representations [52].

3.5.1 Common 2D Metrics

This section presents a brief overview of full-reference quality assessment metrics developed for 2D images and video, since they are typically used as benchmark of the metrics developed for 3D video. The most common 2D metrics are [48] [57]:

- **Peak Signal-to-Noise Ratio (PSNR):** measures the signal accuracy of a distorted image compared to a reference. It is based on the mean squared error (MSE).
- **Structural Similarity (SSIM):** combines image structural information, as pixels mean, variance and covariance, using the full image resolution.
- **Multi-scale SSIM (MSSIM):** applies SSIM over multiple scales of the image, through a process of multiple stages of sub-sampling.
- **Video Quality Metric (VQM):** measures the perceptual effects of video impairments such as blurring, jerky/unnatural motion, global noise, block distortion and color distortion, using a quality metric for each artifact type, and combining all metrics into a single measure.
- **Universal Quality Index (UQI):** models the image distortion by a combination of three factors: loss of correlation, luminance distortion and contrast distortion. It is a perceptual-like method which is an alternative to pixel-based methods.
- **PSNR-HVS:** based on PSNR and UQI, but it takes into account the HVS properties.
- **PSNR-HVSM:** based on PSNR, but it takes into account the contrast sensitivity function (CSF) and between-co-efficient contrast masking of DCT basis function.
- **Visual Signal-to-Noise Ratio (VSNR):** a perceptual-like metric, based on a visual detection of distortion criterion, helped by the CSF.
- **Weighted Signal-to-Noise Ratio (WSNR):** uses a weighting function adapted to the HVS.
- **Visual Information Fidelity (VIF):** uses a statistical model and an image distortion model, on the wavelet domain. VIFP is a pixel-based version of VIP.
- **Information Fidelity Criterion (IFC):** uses a distortion model to evaluate the information shared between the reference image and the degraded image.
- **Noise quality measure (NQM):** quantifies the injected noise in the tested image.
- **Feature Similarity Metric (FSIM):** Although less known, the FSIM [58] uses, as image features, the phase congruency (PC) and the gradient magnitude (GM), exploiting the fact that image positions with high values of PC and GM are those most discernible by the human visual system.

The following sections present an overview of objective metrics that have been recently proposed for the quality assessment of synthesized views using DIBR techniques.

### 3.5.2 Depth Based Perceptual Quality Assessment

**Approach**

Since low depth areas are more vulnerable to rendering distortions than high depth areas, in [53] the authors propose to use a depth-based weighting approach to be applied on a common 2D metric (e.g.: PSNR or SSIM); the weighting penalizes more the distortions occurring in low depth areas. The proposed framework also takes into account the temporal flickering by estimating the temporal consistency among successive synthesized frames. This functionality is achieved by comparing the motion activity in the synthesized video with the one in the reference video.

**Technical Solution**

Figure 3.10 shows the block diagram of the metric proposed in [53]. The reference video and reference depth map represent the viewpoint original information (texture and depth, respectively). The impaired video signal corresponds to the texture component of the synthesized viewpoint.

![Figure 3.10 - Block diagram of the proposed view synthesis quality assessment framework [53].](image)

The objective quality assessment is based on the following key procedures:

1. **Weighting computation based on depth range:** for each frame of the synthesized video, a per-pixel weight, \( WC(u, v) \), is computed using eq. (3.9), where \( Z(u, v) \) is the depth value at pixel position \( (u, v) \) of the reference depth map, and \( Z_f, Z_n \) are,
respectively, upper and the lower threshold depth values; these values correspond to the minimum depth difference between neighboring positions (on the reference view) and between positions that are two pixels apart.

\[
WC(u, v) = \begin{cases} 
0, & Z(u, v) > Z_f \\
\frac{(Z(u, v) - Z_f)}{Z_n - Z_f}, & Z_n \leq Z(u, v) \leq Z_f \\
1, & Z(u, v) < Z_n
\end{cases}
\] (3.9)

2. **Weighting computation based on temporal consistency:** first, the absolute difference between the current time frame and the previous time frame is computed, for the texture component of both original and synthesized views, resulting in \(D_o\) and \(D_s\), respectively. The temporal consistency weighting coefficient is computed using eq. (3.10), where \(RF(u, v) = 1\) represents the risk of flickering at pixel position \((u, v)\) and \(m\) is a motion activity threshold; it considers that motionless frame regions might have suspiciously high motion activity in the synthesized video, causing a flickering effect.

\[
RF(u, v) = \begin{cases} 
1, & D_o(u, v) < m \text{ and } D_s(u, v) > m \\
0, & \text{all other cases}
\end{cases}
\] (3.10)

3. **2D video quality assessment:** an error map is obtained by applying a 2D quality metric (PSNR or SSIM) between each frame of the original and synthesized views.

4. **Computing the final score:** two intermediate quality scores - spatial and temporal - are obtained for each frame; the temporal score results from multiplying the synthesized frame temporal difference (computed using the MSE or the SSIM) with the weighting coefficients given by eq. (3.9), and averaging the resulting error; the spatial score results from multiplying the error map obtained with the 2D metric by the weighting coefficients given by eq. (3.10), and averaging the result. The final score (per frame) results from the spatial and temporal scores average.

**Performance Assessment**

To evaluate the performance of the proposed metric, two different multi-viewpoint video sequences were used: Akko&Kayo and Newspapers. The corresponding depth maps were distorted using four different types of distortion: quantization (during compression), low pass filtering, shifting of the object borders and addition of artificial local spot errors. Also, two different quantization step sizes were used to distort the color texture when compressing the texture component. Table 3.2 shows the Person linear correlation coefficient (PLCC) between the scores resulting from the tested metrics (when applied to the reference and synthesized views), and the scores resulting from the VQM metric. According to this table, the proposed metric shows a better performance than the conventional 2D metrics; however, it is important to note that these results may not reflect the real quality of the techniques, since VQM cannot replace the subjective assessment (MOS values).
Table 3.2 – PLCC of the evaluated quality metrics.

<table>
<thead>
<tr>
<th>Objective Metric</th>
<th>Akko&amp;Kayo</th>
<th>Newspapers</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>0.960</td>
<td>0.884</td>
</tr>
<tr>
<td>PSNR with the proposed method</td>
<td>0.972</td>
<td>0.972</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.505</td>
<td>0.408</td>
</tr>
<tr>
<td>SSIM with the proposed method</td>
<td>0.790</td>
<td>0.507</td>
</tr>
</tbody>
</table>

3.5.3 Quality Assessment Based on Spatial Features

Approach

As in the previously described work, the method proposed in [55] also weights the distortion values obtained from a 2D image quality assessment metric; however, in [55] these weights are computed based on three different maps: texture-based map, orientation-based map and contrast-based map. This is supported by the fact that the perception of artifacts (on the synthesized views) is strongly linked with spatial features, notably with the complexity in terms of textures, the diversity of gradient orientations and the presence of high contrasts.

Technical Solution

Figure 3.11 describes the block diagram of the proposed metric, named by the authors as View Synthesis Quality Assessment (VSQA).

![Figure 3.11 – Block diagram of the quality measurement system proposed in [55].](image)

To compute the VSQA distortion map, the following steps are applied:

1. **Distortion Map (dist):** a 2D quality metric (e.g., SSIM) is applied between original and synthesized views, resulting in a distortion map.

2. **Texture based weighting map (\(W_t\)):** since the perception of artifacts surrounded by high gradient pixels is attenuated due to masking effect, the method starts by computing a texture-based visibility map, \(V_t\), using the Sobel operator. A texture-based weighting map is then created by rescaling the texture-based visibility map between 0 and 2, as described by eq. (3.11); \(\min(V_t)\) and \(\max(V_t)\) are computed over the visibility map, \(V_t\):

   \[
   W_t(u, v) = \frac{2}{\max(V_t) - \min(V_t)} \times V_t(u, v) - \frac{\min(V_t)}{\max(V_t) - \min(V_t)} \quad (3.11)
   \]

3. **Orientation based weighting map (\(W_o\)):** the orientation-based weighting map, \(W_o\), quantifies
the diversity of gradient orientations around each pixel; it aims to model the masking effect due to large diversity of gradient orientations - when an artifact is located in a high gradient orientations diversity area, the weighting decreases the corresponding distortion value (and vice-versa). First, an orientation-based visibility map, \( V_o \), is computed as the standard deviation of the gradient orientations in each pixel neighborhood. The orientation-based weighting map is created by rescaling \( V_o \) between 0 and 2, as described by eq. (3.11), and replacing \( V_t \) by \( V_o \).

4. **Contrast based weighting map (\( W_c \))**: A contrast-based visibility map, \( V_c \), is computed as the average luminance difference between each pixel and the pixels of its neighborhood. If the luminance difference is not significant, the weighting should decrease the corresponding distortion value (and vice-versa). The contrast-based weighting map is created by rescaling \( V_c \) between 0 and 2, as described by eq. (3.11), and replacing \( V_t \) by \( V_c \).

5. **VSQA distortion map**: The VSQA distortion map is computed by eq. (3.12):

\[
VSQA(u, v) = dist(u, v) \times [W_t(u, v)]^\delta \times [W_o(u, v)]^\epsilon \times [W_c(u, v)]^\xi ,
\]

where \( dist(u, v) \) denotes the distortion value given by the chosen 2D metric at pixel position \((u, v)\); \( W_t, W_o, W_c \) correspond, respectively, to the texture, orientation and contrast-based weighting maps; \( \delta, \epsilon \) and \( \xi \) are parameters to adjust the importance of the weighting maps.

6. **VSQA global score**: In order to obtain a global score, a threshold given by eq. (3.13) is applied to the VSQA distortion map:

\[
th = \min_{VSQA} + p \times \frac{\max_{VSQA} - \min_{VSQA}}{100} ;
\]

pixels whose VSQA distortion value is below \( th \) are cleared; the number of remaining pixels after thresholding represents the VSQA score. In eq. (3.13), \( p \) is a positive parameter, and \( \max_{VSQA}, \min_{VSQA} \) are the maximum and the minimum VSQA distortion values, respectively.

Performance Assessment

For the assessment of the proposed metric, three sequences of the IRCCyN/IVC DIBR database [59] were used, namely: *BookArrival, Lovebird1* and *Newspaper*. The synthesized sequences, corresponding to four different viewpoints for each one of the three test videos, were obtained by seven different DIBR methods, resulting in 84 synthesized videos. Table 3.3 presents the PLCC between subjective measurements (DMOS values) of the synthesized views, and the scores resulting from the proposed SSIM-based VSQA and also from conventional 2D quality metrics. These results show that the proposed metric improves the performance of the SSIM metric, and is better than any considered 2D metric.

<table>
<thead>
<tr>
<th>PSNR</th>
<th>SSIM</th>
<th>SSIM-based VSQA</th>
<th>MSSIM</th>
<th>VSNR</th>
<th>VIF</th>
<th>VIFP</th>
<th>UQI</th>
<th>IFC</th>
<th>NQM</th>
<th>WSNR</th>
<th>PSNR HVS</th>
<th>PSNR HVS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.45</td>
<td>0.44</td>
<td>0.62</td>
<td>0.56</td>
<td>0.36</td>
<td>0.32</td>
<td>0.26</td>
<td>0.39</td>
<td>0.28</td>
<td>0.53</td>
<td>0.44</td>
<td>0.41</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Table 3.3 – PLCC between DMOS values and objective quality scores.
3.5.4 Quality Assessment Based on Contours Consistency and Critical Areas

Approach

Artifacts resulting from DIBR algorithms are different from those resulting from compression, and are located in specific areas, namely in edges and disoccluded areas. Taking these facts into account, two simple approaches are proposed in [48]. The first approach assumes that regions may be resized or slightly shift, depending on the chosen extrapolation method (for predicting the disoccluded areas); however, if still visually coherent, MOS values can be high. Accordingly, the method should be able to discriminate between consistent and non-consistent edges distortion and weight the resulting differences (between original and virtual views) accordingly. The second approach consists in applying a 2D metric only on the critical areas. The critical areas are defined by the main differences between the reference view and the synthesized view; the SSIM measure is applied on these areas and the final score is the mean of SSIM scores normalized by the amount of distorted pixels.

Technical Solution

The quality score of the first metric, illustrated in Figure 3.12 (a), is computed by the following steps:

1. **Edge detection**: consists in identifying the position of the edges in both original and synthesized views, by applying a Canny Edge Detector [60].

2. **Motion vector estimation**: motion vectors are estimated over the edges of the synthesized view, in comparison with the edges of the reference view.

3. **Penalization of pixels**: considering a 3×3 neighborhood, neighbor vectors whose angle with the motion vector of the central pixel is superior to 45° are considered as deviant and the central pixel is penalized by voting 1 for each deviant vector. The score of the central pixel is the sum of all votes of the neighbor vectors.

4. **Quality score**: the final score of the image is obtained by computing an average of all pixels votes obtained previously.

![Figure 3.12](image-url)

**Figure 3.12** – Block diagram of the proposed techniques: (a) based on contours distortions analysis; (b) based on critical areas analysis [48].

The quality score of the second metric, illustrated in Figure 3.12 (b), is computed by the following steps:
1. **Difference between views**: the difference between original and synthesized views is computed.

2. **Construction of a mask according to main differences**: in order to detect the critical areas, a threshold $Th$ is applied to the views difference; $Th$ is defined as:

   $$Th = \frac{\max(I - I')}{10}$$

   where $I$ is the original image and $I'$ is the synthesized image.

3. **SSIM**: the SSIM measure is applied to the critical areas obtained in step 2.

4. **Quality score**: the final quality score of the synthesized image is the mean of SSIM scores normalized by the amount of distorted pixels (pixels belonging to the critical areas).

**Performance Assessment**

To assess the performance of the proposed metrics three test sequences (*Book Arrival*, *Lovebird1* and *Newspaper*) were used to generate four different viewpoints, which led to twelve images. Then seven different DBIR algorithms were used to synthesize the original images. In total, 84 synthesized images were used in the performance assessment. In this experience PLCC was used to correlate the objective and subjective measures. For each proposed metric, the PLCC was computed between the respective metric and MOS and it was compared with PLCC between SSIM and MOS. The results for the first technique showed that the PLCC were 0.18 and 0.84 between the objective metrics (SSIM and the first technique) and subjective metric (MOS), respectively. The results for the second technique showed that the PLCC were 0.18 and 0.78 between the objective metrics (SSIM and the second technique) and subjective metric (MOS), respectively.

### 3.5.5 Quality Assessment Based on SSIM and Non-consistent Edges Shifts

**Approach**

As mentioned before, object shifts on the synthesized views may result in high pixel differences (between synthesized and original views) although not affecting the perceived quality. Accordingly, to properly assess the quality, the authors of [56] suggest to first compensate the “consistent” object shifts before applying a conventional 2D quality metric (e.g., SSIM), which will result in a "quality score". The technique proposed in [56] also computes a "structural score", using edge detection and applying the Hausdorff distance [61] between reference and synthesized images edges; the goal is to measure the ghost-type distortions or inconsistent objects shifts. The final quality score is obtained after pooling the two quality results.

**Technical Solution**

Figure 3.13 presents the block diagram of the quality metric proposed in [56].
Figure 3.13 – Block diagram of the metric proposed in [56].

The quality score \( Q \) is obtained by the following steps:

1. **NxN blocking**: both original and distorted images are split in \( N \times N \) disjoint blocks.

2. **Shift Compensation**: objects shift compensation is performed using a block-matching algorithm: for each block in the synthesized image, the best corresponding block in the original image is found.

3. **Gaussian filter**: both original and synthesized views are smoothed with a Gaussian filter, with the purpose of deleting pixel errors with low subjective impact.

4. **Obtain the quality score \( Q(D) \)**: a conventional 2D quality model (e.g., SSIM) is applied to each pair of matching blocks from the original and the synthesized images.

For computing the structural score, and after splitting the images in \( N \times N \) disjoint blocks and performing the shift compensation as explained above, the following steps are applied:

1. **Edge detection**: edges are detected by applying the Canny edge detector [60] to the original and synthesized images.

2. **Hausdorff distance \( H(D) \)**: to measure the degree of edges distortion (e.g., ghost-type artifacts) the Hausdorff distance is computed between the edges of the original and synthesized views, using each pair of matching blocks.

3. **Obtain the structural score \( S(D) \)**: this score is given by eq. (3.15):

   \[
   S(D_i) = 1 - H_{normalize}(D_i),
   \]

   where \( H_{normalize}(D_i) \) is the normalized Hausdorff distance for the \( i^{th} \) block.

Before computing the final score of the synthesized image, the quality and structural scores are combined by eq. (3.16), where \( F(D_i) \) is the final score for the \( i^{th} \) block and \( \alpha \) denotes the importance of each score:

\[
F(D_i) = \alpha \cdot Q(D_i) + (1 - \alpha) \cdot S(D_i)
\]

The final score results from the average of the \( p\% \) lowest \( F(D_i) \) scores, according to eq. (3.17), where \( N_p \) is the corresponding number of blocks:

\[
final\_score = \frac{1}{N_p} \sum_{n}^{N_p} F(D_n)
\]
Performance Assessment

In order to assess the performance of the proposed metric, six multiview plus depth sequences provided by MPEG 3DVC for the 3DVC contest were used (Poznan Hall2, Kendo, Balloons, Poznan Street, Lovebird, and Newspaper1). Three types of distortion were applied to the depth maps: Offset, Quantization and Gaussian noise. Offset distortion was created by adding a constant value to all the pixels of the depth map. Quantization distortion consisted on quantizing the depth map values into specific levels. Gaussian noise added white Gaussian noise, with different variances, to the depth map. In the experiment, two cases were considered: Case 1 includes the depth maps with Offset and Quantization distortions; Case 2 includes the three types of distortions. The results presented in Table 3.4 and Table 3.5 show that, in comparison with the popular 2D models, the proposed method has a higher PLCC, and lower Root Mean Square Error (RMSE) and Outlier Ratio (OR), when matching the subjective scores (MOS values).

<table>
<thead>
<tr>
<th>Metric</th>
<th>PLCC</th>
<th>RMSE</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>0.58</td>
<td>0.70</td>
<td>0.042</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.37</td>
<td>0.80</td>
<td>0.083</td>
</tr>
<tr>
<td>MSSIM</td>
<td>0.51</td>
<td>0.74</td>
<td>0.083</td>
</tr>
<tr>
<td>UQI</td>
<td>0.40</td>
<td>0.79</td>
<td>0.167</td>
</tr>
<tr>
<td>VIF</td>
<td>0.53</td>
<td>0.73</td>
<td>0.083</td>
</tr>
<tr>
<td>VSNR</td>
<td>0.35</td>
<td>0.81</td>
<td>0.167</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.90</td>
<td>0.38</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>PLCC</th>
<th>RMSE</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>0.52</td>
<td>0.71</td>
<td>0.086</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.42</td>
<td>0.75</td>
<td>0.057</td>
</tr>
<tr>
<td>MSSIM</td>
<td>0.42</td>
<td>0.75</td>
<td>0.057</td>
</tr>
<tr>
<td>UQI</td>
<td>0.36</td>
<td>0.77</td>
<td>0.114</td>
</tr>
<tr>
<td>VIF</td>
<td>0.56</td>
<td>0.69</td>
<td>0.029</td>
</tr>
<tr>
<td>VSNR</td>
<td>0.25</td>
<td>0.80</td>
<td>0.200</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.77</td>
<td>0.53</td>
<td>0.029</td>
</tr>
</tbody>
</table>

3.5.6 Quality Assessment Based on Skin Detection

Approach

As in the previous work, the metric proposed in [62] also assumes that objects shifts can be introduced by the rendering process without affecting the visual quality of the synthesized images. Additionally, it considers that humans are more sensitive to artifacts affecting regions containing human beings (i.e., faces or hands), so that distortions on those regions may severely impact the perceived quality. To cope with this fact, the proposed method includes a weighting function based on the results of a skin detection procedure.

Technical Solution

Figure 3.14 presents the block diagram of the method proposed in [62], named by the authors as 3DSwIM (3D Synthesized view Image Quality Metric).
This method relies on a comparison of statistical features, extracted from the wavelet domain, of the original image and synthesized images. The final score of 3DSwIM is achieved by the following steps:

1. **Block partition**: in this step, both original and synthesized images are split into non-overlapping blocks of size $n \times m$.

2. **Registration**: to correct objects shifts, an exhaustive search (ES)-like algorithm [63] is applied between the blocks of the original and synthesized images.

3. **Skin detection**: this procedure detects the image regions that are likely to contain human faces; it is based on a color segmentation process and uses the H component of the HSV (Hue, Saturation and Value) color space [64].

4. **Application of Discrete Wavelet transform (DWT)**: each block from the original and synthesized views goes through a first level Haar wavelet transform [65]. This will allow to measure the image degradation by analyzing the statistical variations in the wavelet sub-band related with the horizontal details; in fact, artifacts in the filled holes from the DIBR process are mainly characterized by high frequencies in the horizontal direction.

5. **Histogram computation**: computes the histogram of the original and of the synthesized blocks wavelet coefficients, and for the sub-band considered in step 4.

6. **KS (Kolmogorov-Smirnov) distance computation**: applies the KS distance to the histograms obtained in step 5.

7. **Block distortion**: computes the block distortion, $d_b$, using eq. (3.18):

$$d_b = \max(|F_{ob} - F_{sb}|) \quad (3.18)$$

where $F_{ob}$ and $F_{sb}$ are the block distribution function of the real and synthesized view, respectively.

8. **Image distortion**: the image distortion, $d$, is computed by eq. (3.19) if the presence of human skin is not taken into account, otherwise is computed by eq. (3.20).

$$d = \frac{1}{D_0} \sum_{b=1}^{B} d_b \quad (3.19)$$

$$d = \frac{1}{D_0} \sum_{b=1}^{B} d_b W_{\text{skin}} \quad (3.20)$$

where $D_0$ is a normalization constant, $B$ is the number of blocks in the image and $W_{\text{skin}}$ is the skin weighting, which is set to 15 if skin is detected in a block.

9. **Image quality score**: the final image quality score, $s$, is computed by eq. (3.21), and may take a value in the interval $[0,1]$.

$$s = \frac{1}{1 + d} \quad (3.21)$$
Performance Assessment

To assess the performance of the proposed metric, three different MVD sequences (Book Arrival, Lovebirds and Newspapers) were used; from each MVD sequence, four different intermediate viewpoints were generated using seven different DIBR algorithms. In total, 84 synthesized views were considered. Table 3.6 presents the PLCC and RMSE between objective scores (for the considered objective metrics) and DMOS values, of the synthesized views; Table 3.7 presents the results of the proposed metric, with and without the use of the skin weighting. These results show that the proposed method outperforms the 2D conventional metrics, and also that skin detection improves the correlation of the objective quality score with the subjective scores.

Table 3.6 - PLCC and RMSE between DMOS and objective scores.

<table>
<thead>
<tr>
<th>Metric</th>
<th>3DSwIM</th>
<th>VSQA</th>
<th>PSNR</th>
<th>SSIM</th>
<th>SSIM</th>
<th>MSSIM</th>
<th>VSNR</th>
<th>UQI</th>
<th>PSNR-HVSM</th>
<th>PSNR-HVS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLCC (%)</td>
<td>0.76</td>
<td>0.54</td>
<td>0.47</td>
<td>0.41</td>
<td>0.55</td>
<td>0.36</td>
<td>0.19</td>
<td>0.43</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>RMSE</td>
<td>0.42</td>
<td>0.58</td>
<td>0.61</td>
<td>0.65</td>
<td>0.59</td>
<td>0.65</td>
<td>0.66</td>
<td>0.63</td>
<td>0.64</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.7 – PLCC and RMSE between DMOS and objective scores.

<table>
<thead>
<tr>
<th>Metric</th>
<th>3DSwIM (full) PLCC</th>
<th>3DSwIM without skin detection PLCC</th>
<th>3DSwIM without skin detection RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book Arrivals</td>
<td>0.97</td>
<td>0.95</td>
<td>0.54</td>
</tr>
<tr>
<td>Lovebirds</td>
<td>0.54</td>
<td>0.35</td>
<td>0.44</td>
</tr>
<tr>
<td>Newspaper</td>
<td>0.68</td>
<td>0.81</td>
<td>0.39</td>
</tr>
</tbody>
</table>

3.5.7 Perceptual Quality Assessment of 3D Synthesized Images

Approach

Since the synthesized images, resulting from DIBR algorithms, have to cope with depth errors, occlusions, imprecise camera parameters and re-interpolation, a novel metric is proposed in [66] to evaluate the quality of synthesized images, named: DIBR-Synthesized image Quality Metric (DSQM). This proposed metric uses a block-based perceptual feature matching based on the signal phase congruency metric to estimate the synthesis distortion.

Technical Solution:

The proposed “DIBR-Synthesized image Quality Metric (DSQM)” is a block-based algorithm that works in several steps which are presented in Figure 3.15:

![Figure 3.15 – Block diagram for DSQM metric.](image)

The final score for DSQM is achieved by the following steps:

1. **Block partition:** in this step, the reference image is split into small blocks of size $m \times n$. 
2. **Image Block Matching**: For each block of the reference image, the best corresponding block in the synthesized image is found, using the disparity information or through a conventional displacement estimation technique (e.g., block matching algorithm) in case the disparity information is unavailable.

3. **Perceptual Feature Extraction**: In this step, perceptual features are extracted from the matching block pairs; the used feature is the mean value of the Phase Congruency (PC) map \( \mu(PC) \).

4. **Compute Image Block Distortion**: The synthesis distortion, \( Q_i \), corresponding to block \( i \) of the reference image is measured by computing the absolute difference between the mean value of the phase congruency maps for that block and for the corresponding block, \( j \), on the synthesized image, according to (3.22)

\[
Q_i = |\mu(PC_i) - \mu(PC_j)| .
\] (3.22)

5. **Compute Overall Image Quality**: The overall quality score for the synthesized image is computed by averaging the quality scores of every matching blocks, according to (3.23)

\[
DSQM = \frac{1}{k} \sum_{i=1}^{k} Q_i .
\] (3.23)

**Performance Assessment**:

To assess the performance of the proposed metric, the IRCCyN database, previously described in section 3.5.6, was used. The quality of each synthesized image was estimated using, as reference, the view from which the synthesized view was generated (i.e., the source view). Table 3.8 presents the PLCC, SROCC and RMSE between the objective scores and the DMOS values of the synthesized views, for the proposed metric and other existing metrics. The results show that the DSQM outperforms the well-known 2D metrics as well as the 3DSwIM metric, previously overviewed.

<table>
<thead>
<tr>
<th></th>
<th>DSQM</th>
<th>3DSwIM</th>
<th>SSIM</th>
<th>MS-SSIM</th>
<th>PSNR</th>
<th>VSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLCC</td>
<td>0.7895</td>
<td>0.6429</td>
<td>0.5639</td>
<td>0.5489</td>
<td>0.4283</td>
<td>0.5145</td>
</tr>
<tr>
<td>SROCC</td>
<td>0.7151</td>
<td>0.5613</td>
<td>0.4687</td>
<td>0.5324</td>
<td>0.4628</td>
<td>0.5051</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.4086</td>
<td>0.5105</td>
<td>0.5499</td>
<td>0.5566</td>
<td>0.6017</td>
<td>0.5709</td>
</tr>
</tbody>
</table>

*Table 3.8 – PLCC, SROCC and RMSE between DMOS and objective scores [66].*
Chapter 4. Proposed Models for Quality Assessment of Synthesized Images

4.1 Introduction

In the previous chapters, the most relevant blocks of a 3D video transmission chain were reviewed, with focus on the multiview plus depth 3D video representation, especially targeting free viewpoint applications. In these applications, the synthesis of intermediate viewpoints is usually made with a Depth Image Based Rendering (DIBR) technique, which may lead to specific image artifacts due to the lack of accuracy in the depth acquisition and/or depth map compression, and also due to the view synthesis process limitations (especially in occluded areas). However, the impact of these artifacts on the user perceived quality cannot be properly assessed with conventional 2D quality metrics. Accordingly, specific quality metrics for synthesized views have been recently proposed in the literature and some of them were reviewed in the previous chapter. Most of the techniques rely on identifying the areas where the synthesis artifacts are more likely to occur, such as object edges, and/or where these artifacts have a higher impact in the human perception, such as lower depth areas and human skin. However, despite the relevance of edges distortions on the synthesized images quality, their impact on the perceived image quality was still not properly addressed in the literature.

In this chapter, a full reference (FR) quality assessment model for DIBR based synthesized images is proposed. Inspired by the work of [56], this model also combines two metrics: a conventional 2D image quality metric, which main propose is to account for the perceptual impact of artifacts resulting from common image processing procedures (e.g., compression), and a structural metric, to account for synthesis specific artifacts. Comparatively to [56], an in-depth evaluation of several edges distortion metrics was conducted, resulting in a new FR model with higher performance.

Section 4.2 presents the approach followed in the proposed quality assessment model for synthesized images; sections 4.3 and 4.4 detail the main model components, namely, 2D Image Quality Metric and Structural Quality Metric; section 4.5 describes the fusion procedure of the aforementioned metrics, that results in a unique and global image quality score.

4.2 Approach

The proposed model, which block diagram is depicted in Figure 4.1, is composed by two main modules: 2D Image Quality Metric and Structural Quality Metric. The 2D Image Quality Metric module aims to predict the impact, on the perceived image quality, of the artifacts resulting from common image processing procedures (e.g., compression); the objective of the Structural Quality Metric module is to assess the impact of edges distortions.

Each module receives two images as input: a reference view and the synthesized view, which quality should be assessed. The reference view can be the original image corresponding to the synthesized view, resulting in a quality metric typically refereed as “full reference”, or one of the lateral views used for the synthesis process. Both metrics are block based, i.e., each metric results in a quality score per image block. These scores are then combined by the Fusion module, resulting in a
single and global quality score for the synthesized view. The main motivation behind the use of a block based approach is to better detect the synthesis artifacts, since their distribution on the image is quite uneven.

![Block Diagram of the Proposed Quality Metric](image1)

**Figure 4.1 – Block Diagram of the Proposed Quality Metric.**

The following sections describe the solutions designed and implemented for each module of Figure 4.1, highlighting the rationale behind them. It is important to note that two solutions will be proposed for the structure quality metric, one based on the Hausdorff distance and other based on the image gradient.

The assessment of each solution, and of the entire proposed model, is provided in the next chapter.

### 4.3 2D Image Quality Metric

Figure 4.2 illustrates the block diagram of the 2D Image Quality Metric module.

![Block Diagram of the 2D Image Quality Metric](image2)

**Figure 4.2 – Block Diagram of the 2D Image Quality Metric.**

Since the main objective of the 2D Image Quality Metric is to quantify the artifacts resulting from common image processing techniques (e.g., compression, noise reduction filtering), any conventional full-reference quality metric developed for that purpose (e.g., SSIM, FSIM, PSNR) can be potentially used. These metrics require that both original and distorted images be spatially synchronized; accordingly, before applying them, consistent shifts introduced by the synthesis process need to be estimated and compensated. As mentioned previously, this compensation is necessary to guarantee that the quality prediction is not penalized due to those shifts which do not affect the quality of experience. An example of this situation is presented in Figure 4.3, where an horizontal displacement of 44 pixels was identified by comparing the coordinates of corresponding points in the original and synthesized images; yet, the quality of the synthesized image in the neighborhood of the signalized points is quite good.
The consistent shifts are estimated by computing the displacement vectors between the reference and synthesized images, using a traditional block-based motion estimation technique. These vectors are then applied to the synthesized image to produce a shift compensated texture image, spatially synchronized with the reference one.

To estimate the displacement vectors, the block matching algorithm with Exhaustive Search (BM-ES) was considered. Although being more computationally demanding than sub-optimum search procedures (e.g., Four Step Search, Diamond Search, Three Step Search, New Three Step Search, Adaptive Rood Pattern Search), the resulting PSNR between the compensated texture and the synthesized one is higher; also, real time implementation of the metric was not a target of this work.

To increase the reliability of the estimated displacement vectors, a hierarchical BM-ES procedure was implemented: in a first iteration, the BM-ES is applied with large block sizes (e.g. 128x128 pixels) and a large search window (e.g., 50 pixels), which are progressively decreased in the following iterations. Figure 4.4 illustrates the search scheme, with the following definitions:

- \( p_1 \) – maximum search displacement on the vertical direction (below and above the current block);
- \( p_2 \) – maximum horizontal displacement on the horizontal direction (to the right and left sides of the current block);
- \( mbSize_1 \) – vertical block size;
- \( mbSize_2 \) – horizontal block size.

The parameters \( p_1 \) and \( p_2 \) are defined independently to have a more flexible window search size. Assuming that the image/video acquisition cameras are parallel (which is typically the case) the disparity is null on the vertical direction, resulting that \( p_1 \) can be considered close to 0.
After applying the first iteration of the hierarchical BM-ES, each block will have an associated displacement vector; in the next iteration, the search window will be centered on the position pointed by that vector, and this procedure is also applied in the following iterations (i.e., each iteration has a search window centered on the position pointed by the displacement found till the end of the previous iteration).

The motion vectors estimated by hierarchical BM-ES are more reliable than using a non-hierarchical method, since starting with large blocks reduces the occurrence of local minima.

The implemented hierarchical BM-ES has three iterations and, at each iteration, the block size is halved in each direction. The final displacement vector for each block results from the sum of the displacement vectors obtained in each iteration, as illustrated in Figure 4.5: for the block on the left side, the estimated displacement after the first iteration is $\mathbf{v}_1$; this block is split in four blocks in the second iteration, and the estimated displacement for the upper-left block is $\mathbf{v}_2$; in the third iteration, the four blocks of the second iteration are split again, resulting in sixteen blocks and in vector $\mathbf{v}_3$ for the upper-left block. The final displacement vector for the block signalized in green in Figure 4.5 is then given by $\mathbf{v} = \mathbf{v}_3 + \mathbf{v}_2 + \mathbf{v}_1$.

After displacement estimation, the consistent shifts of the synthesized image are then compensated according to the found displacement vectors.

Finally, the quality of the compensated synthesized image is then evaluated using a 2D full-reference quality metric. This procedure is applied at the block level and the 2D metrics considered are: MSE (mean squared error), SSIM and FSIM.
4.4 Structural Quality Metric

The Structural Quality Metric module aims to quantify edges distortions; comparatively to the 2D module, its focus is the objects structure, with the goal of measuring the objects deformations. To quantify the edges distortions, two approaches have been studied, the first one based on the Hausdorff distance [68] and the second one based on the histogram of the image gradient.

4.4.1 Hausdorff Distance Based Approach

The Hausdorff distance (HD) based approach targets to detect and quantify non-consistent edge shifts and edges deformations, which may have a negative impact on the QoE. It is based on the classical HD, but some modifications were proposed that improved the performance. Figure 4.6 depicts the general block diagram of the HD based approach; its main building modules are detailed in the following text.

![Figure 4.6 – Block diagram of the Hausdorff based approach.](image_url)

A. Shift Estimation and Compensation

The HD is computed between the edges of the reference and synthesized images; however, as for the 2D Quality Metric, consistent shifts need to be compensated beforehand. For this, the displacement estimation algorithm described in section 4.3 is also used. However, since the HD relies on images edges, different approaches were considered regarding the input data for the displacement estimation and compensation, which are described below:

**S1:** displacement vectors are estimated using the texture images, and the edge detector is applied to the reference image and to the synthesized image after shift compensation;

**S2:** displacement vectors are estimated using the texture images, and the edge detector is also applied to those images. The shift compensation is only applied to the edge map of the synthesized image and the other edge map is kept unchanged;

**S3:** the edge detector is applied to the texture images, and the displacement estimation uses the edges map; the shift compensation is only applied to the edge map of the synthesized image and the other edge map is kept unchanged;

After a first experimental evaluation of the three solutions, both S1 and S3 were discarded. In the S1 case, applying the edge detector to the texture images after shift compensation introduces false
edges, resulting from the borders of shifted blocks. In the S3 case, applying the displacement estimation on the image edge maps results in unreliable estimates, and the shift compensation would not perform so well. Therefore, the solution S2 was chosen since it applies the displacement estimation to the texture images, resulting in more robust estimates. The shift compensation is applied to the image edges, minimizing the risk of creating false edges. Figure 4.7 depicts the block diagram of solution S2.

![Figure 4.7 – Block Diagram of the shift estimation and compensation.](image)

**B. Edges Detection**

Edges detection is a non-trivial task, and several edge detections methods have been proposed in the literature, showing different levels of complexity and success rates. Most of them start by computing the image gradient, following by the selection of the most evident edges, which occur when the gradient magnitude is greater than a specific threshold.

In this work, edges are detected using the Canny edge detector [60], available in Matlab; it is one of the most powerful edge detectors, in terms of edge detection rate, edge location and noise robustness. The method starts with a Gaussian filtering, to smooth the noise on the image, and the gradient is computed for each image pixel. The gradient magnitude is then assessed, to classify each pixel as an edge or non-edge position. The algorithm uses two different thresholds, \( T_1 \) and \( T_2 \), with \( T_1 < T_2 \). Points with gradient value above \( T_2 \) are considered as "strong edges", whereas points with gradient below \( T_1 \) are considered "non-edges"; points with a gradient between \( T_1 \) and \( T_2 \) are considered "weak edges". Finally, the method performs edge linking, which considers as edges, "weak edges" that are 8-connected to a "strong edge". The thresholds can be adapted to the image content, using the Otsu’s Method [69], or a priori selected. To ensure that the edge detection is properly achieved, i.e., being able to identify strong edges and minimize the occurrence of false edges, the impact of the choice of the thresholds was evaluated, and will be described on the next chapter.

**C. Hausdorff Distance**

Let consider two set of points, \( A = \{a_1, \ldots, a_N\} \) and \( B = \{b_1, \ldots, b_N\} \). The Hausdorff distance (HD) measures how far the two sets are from each other. Assuming that the distance between two points \( a \) and \( b \) is defined as the Euclidean distance \( d(a,b) = \|a-b\| \), the distance between a point \( a \) and a set of points \( B = \{b_1, \ldots, b_N\} \) is defined as \( D(a,B) = \min_{b \in B} \|a-b\| \). The common Hausdorff distances, proposed by Huttenlocher [70], are presented from (4.1) to (4.6).

\[
h_1(A,B) = \min_{a \in A} D(a,B) \tag{4.1}
\]
\[ h_2(A, B) = \frac{50K_{th}}{a \in A} d(a, B) \]  
\[ h_3(A, B) = \frac{70K_{th}}{a \in A} d(a, B) \]  
\[ h_4(A, B) = \frac{90K_{th}}{a \in A} d(a, B) \]  
\[ h_5(A, B) = \max_{a \in A} d(a, B) \]  
\[ h_6(A, B) = \frac{1}{Na} \sum_{a \in A} d(a, B) \]  

In this Thesis some variations for Hausdorff distance were also proposed, that are described by (4.7) to eq. (4.9).

\[ h_7(A, B) = \sum_{a \in A} d(a, B) \]  
\[ h_8(A, B) = \sum_{a \in A} H(d(a, B), \delta) \]  
\[ h_9(A, B) = \sum_{a \in A} d(a, B) \times H(d(a, B), \delta) \]

where \( H(.) \) is the Heaviside function, defined by eq. (4.10), and \( \delta \) is a distance threshold.

\[ H(d(a, B), \delta) = \begin{cases} 
1, & d(a, B) > \delta \\
0, & d(a, B) < \delta 
\end{cases} \]  

Equations (4.7), (4.8) and (4.9) are quite different from the common Hausdorff distances since they do not exactly return a distance. In fact, (4.7) returns a sum of all the distances between a point \( a \) and a set of points \( B = \{b_1, ..., b_N\} \) and (4.8) and (4.9) are more complex since they introduce a new variable \( \delta \), which is a distance threshold. The introduction of the \( \delta \) threshold aims to only take into account the distances which are bigger than a certain value, this approach expects that the smaller pixels shifts do not affect the users’ perception of quality and taking into account this shifts could jeopardize the metric results. Equations (4.8) calculates the number of pixels with a displacement higher than \( \delta \) and (4.9) sums the distance of all the pixels that have a displacement higher than \( \delta \). Since the Hausdorff distance is asymmetric, several possibilities can be used to combine the distance from set \( A \) to set \( B \) \((h(A, B))\) and the distance from set \( B \) to set \( A \), \((h(B, A))\), represented as \( H(A, B) \). Equations (4.11) to (4.14) describe the most common \( H(A, B) \) combinations.

\[ H_4(A, B) = \min(h(A, B), h(B, A)) \]  
\[ H_5(A, B) = \max(h(A, B), h(B, A)) \]  
\[ H_6(A, B) = \frac{h(A, B) + h(B, A)}{2} \]  
\[ H_7(A, B) = \frac{Na h(A, B) + N_b h(B, A)}{Na + N_b} \]

In the context of this Thesis the sets \( A \) and \( B \) correspond, respectively, to the edge pixels of the synthesized image and to the edge pixels of the reference image. Figure 4.8 a), b) and c) show, respectively, an original image, the corresponding synthesized image (i.e., same view) and the synthesized image with shift compensation. The corresponding edges maps are depicted in Figure 4.8
d), e) and f). Note that, as explained before, Figure 4.8 f) was obtained through shift compensation of Figure 4.8 e).

Figure 4.8 g) and h) show, respectively, the resulting overlapped between the edges pixels of the original and synthesized images, without and with displacement compensation. Since the edge maps are represented in the same image, shifts and distortions can be identified visually. The grey background represents pixels where no edges were detected or where both edges maps coincide. Comparing Figure 4.8 g) and h), it is clearly visible that when shift compensation is applied the differences between original and synthesized images become much less evident. The comparison of these two figures shows the importance of applying the shift compensation, since most edges differences result from consistent shifts (which are perceptually irrelevant) and not from truly edges distortions.

The Hausdorff distance is then computed using, as set $A$ the image edges of synthesized image after shift compensation (depicted in Figure 4.8 f)) and using as set $B$ the edge map of reference image (depicted in Figure 4.8 d)).

Figure 4.8 – a) original Image; b) Synthesized image; c) Synthesized image after shift compensation; d) Original image edges; e) Synthesized image edges; f) Synthesized image edges after shift compensation; g) overlap of d) and e; h) Overlap of d) and f).
To find for each point of set $A$, the nearest point of set $B$, $D(a,B)$, the \textit{knnsearch} function, available in Matlab, was used. A zero matrix with the same size as the image, is firstly created, and for the pixels (and matrix) positions where an edge is found, the matrix will store the value of $D(a,B)$. Then, the pixels where no edges are found the value of -1 is attributed. Since the Hausdorff distance is asymmetric, the same procedure is applied to find the distance between set $B$ and $A$. Thus, at the end, two matrices are obtained, one with the nearest distance between each edge pixel of $A$ and the edges of $B$ and another with the nearest distance between each edge pixel of $B$ and the edges of $A$.

After obtaining the matrices of distances, each matrix is split into non-overlapped blocks (with the same size of the blocks at the last hierarchical level of the displacement estimation procedure) and the Hausdorff distance is computed per block. Figure 4.9 a) shows an example of a block picked from Figure 4.8 g) and its respective matrix of distance from set $A$ to set $B$, depicted in Figure 4.9 b). The pixels from Figure 4.9 b) which are colored in green express $D(a,B)$. In each block, $h(A,B)$ uses $D(a,B)$ values to calculate the distance from set $A$ to set $B$, and the same happens for $h(B,A)$. In this process only the non-negative numbers are considered, since the pixels with the value -1 represent pixels where no edges were found. Finally, $H(A,B)$ combines the distances of $h(A,B)$ and $h(B,A)$, obtaining thus the Hausdorff distance for the respective block. Note that in this particular example of Figure 4.9, no coincident edges existed, so there is no pixels with 0 distance colored with green.

![Figure 4.9 - Distance between sets: a) Edge block extracted from Figure 4.8 g); b) Matrix containing the distances from set $A$ to set $B$, from the extracted block.](image)

After obtaining the Hausdorff distance for a block, the result is normalized, ensuring that the structural score has a value between 0 and 1. Taking into account the variations of $h(A,B)$, three different output results are achieved:

- **From $h_1$ to $h_6$:** these equations return a value which correspond to a single distance. Based in [56], the considered normalization factor is the block width.

- **$h_7$ and $h_9$:** these two equations return a sum of distances so, in this case, a different approach is necessary. To reach the normalization factor, the histogram corresponding to number of occurrences for each result in the IRCCyN database, which will be introduced in the next chapter, was analyzed. An example is depicted in Figure 4.10. By inspecting the
histogram, different values (e.g. 200, 300, 400, etc.) are tested and the one who gives better results is defined as normalization factor.

- $h_b$: this equation returns a sum of points for which the resulting distance is greater than a certain threshold. Similarly with $h_7$ and $h_9$, the normalization factor was also reached by inspecting the histogram - an example is also illustrated by Figure 4.11, which leads to different possible normalization factors (e.g. 10, 15, 20, etc.).

Thereafter, and depending of the normalization factor, the structural score, $S$, can be higher than 1. In this case, the structure score value is truncated at 1, (4.15) is used:

$$S_{normalized} = \min(H(A,B), 1)$$  \hfill (4.15)

To make 0 corresponding to the worst quality and 1 to the best quality, (4.16) is applied.

$$S = 1 - S_{normalized}$$  \hfill (4.16)

Concluding, a final structural score, per image block, resulting from the Hausdorff distance based approach was obtained in this step.
### 4.4.2 Gradient Based Approach

The gradient based approach uses the histogram of the gradient to assess the edge distortions. This approach computes the histogram of the gradient for the reference and for the synthesized images which are then compared to obtain a structural score. Figure 4.12 presents the block diagram of gradient based approach, which main modules are detailed in the following text.

![Figure 4.12 – Block diagram of Gradient based approach.](image)

#### A- Consistent Shifts Estimation

As for the HD based approach, consistent shifts between reference and synthesized images should be firstly estimated and compensated. The displacement vectors are estimated by applying the hierarchical BM-ES algorithm described in section 4.4.1, to both texture images; the gradient of the synthesized image is then compensated using the estimated vectors.

#### B- Gradient Calculation

Being $I$ the intensity of a digital image (i.e., any color component from the used colors space) and $I(i,j)$ its value in position $(i,j)$, the gradient of $I$ is obtained by applying the differential operator to image $I$, expressed by (4.17).

$$\nabla I(i,j) = \left( I_x(i,j), I_y(i,j) \right)$$

(4.17)

where, $I_x(i,j) = \frac{\partial I(i,j)}{\partial x}$ and $I_y(i,j) = \frac{\partial I(i,j)}{\partial y}$.

The gradient magnitude and phase are obtained through (4.18) and (4.19), respectively:

$$|\nabla I(i,j)| = \sqrt{\left(I_x(i,j)\right)^2 + \left(I_y(i,j)\right)^2}$$

$$\theta(i,j) = \tan^{-1} \frac{I_y(i,j)}{I_x(i,j)}$$

(4.18) 

(4.19)

In the discrete domain, the gradient operator is typically implemented through a convolution of the image with a 2D filter - in this Thesis, the Sobel filter was used.

#### C- Histogram Computation

The histogram computation was conducted using the histogram of oriented gradients (HOG). The HOG counts the occurrences of gradient orientation in an image - each pixel within the image contributes with a weighted vote for an orientation-based histogram based on the value of its gradient. The histogram bins, corresponding to the gradient orientation, are evenly spread over 0 to 180
degrees or 0 to 360 degrees, depending on whether the gradient is “unsigned” or “signed”. In this thesis the convention of “signed” was used to distinguish opposite directions; given the example that two opposite faces of a square have the same direction, one distorted and the other non-distorted, using the “unsigned” convention the two faces will be assigned to the same bin but using the “signed” convention the two faces will be differentiated resulting in a better assessment of distortions. As for the vote weight, pixel contribution can either be the gradient magnitude itself, or some function of the magnitude. In this Thesis, two vote weights were considered:

- **the pixel gradient magnitude** - in this case, the amplitude of the histogram bin corresponding to the orientation \( \theta \) results from the sum of the gradient magnitudes of every pixel with gradient phase \( \theta \);
- **an unit vote** - in this case, the amplitude of the histogram bin corresponding to the orientation \( \theta \) is the number of pixels with gradient phase \( \theta \).

To reduce the contribution of the image noise to the HOG computation, only pixels with gradient magnitude higher than 75 were considered; this value was defined by inspecting the gradient magnitude values in smooth areas of the image.

Although the image gradient is computed for the whole image, the HOG can be computed per block and, in this case, the histograms computation will be made at the block level.

To compute the image gradient, the Matlab function `imgradient` was used; since this function returns the gradient direction in a range from \(-180^\circ\) to \(180^\circ\), a value of \(360^\circ\) was summed to every negative orientation angle, resulting in histogram bins from 0 to \(360^\circ\). To compute the HOG, 16 bins were considered, therefore each bin has a size of \(22.5^\circ\), which seemed a reasonable interval to differentiate directions; Figure 4.13 illustrates the correspondence between the bin number and the gradient orientation angle.

![Figure 4.13 – Illustration of bin disposition to compute the HOG.](image)

**D- Histogram Comparison**

After computing the HOG for reference and synthesized views, the resulting histograms are compared on the “Histogram Comparison” module. In this step, different histogram comparisons algorithms were considered. To ensure that all the following metrics return a value between 0 and 1, the block-based
HOG of the reference and synthesized images were normalized by its respective area. The considered histogram comparisons algorithm are the following:

- **Structural component of edge-based structural similarity (ESSIM)** [71], histogram comparison, $e$: the ESSIM is based on SSIM and aims to compare the edge information between the synthesized image and the reference one. The structural component, $e$, results from comparing the histograms of gradient from the synthesized and the reference images, which can be obtained by (4.20), where $x$ and $y$ represent the block-based HOG of the synthesized and reference images, respectively, $\sigma_x$ and $\sigma_y$ are the standard deviation of $x$ and $y$, and $\sigma_{x,y}$ is the covariance between $x$ and $y$. In this equation, $c$ is a small value to prevent a division by zero.

$$e = \frac{\sigma_{x,y} + c}{\sigma_x \sigma_y + c}$$  \hspace{1cm} (4.20)

- **Kolmogorov-Smirnov (KS)**: this metric is computed according to (4.21), where $CF_x$ and $CF_y$ represent the cumulative function of the block-based HOG of the synthesized image and of the reference one respectively.

$$KS = \max \left( |CF_x - CF_y| \right)$$  \hspace{1cm} (4.21)

- **Bhattacharyya (BC)**: this technique measures the amount of overlap between two histograms and is formulized by (4.22), where $x_i$ and $y_i$ are the bins amplitude of the block-based HOG of the synthesized and the reference images, respectively; In (4.22) and until (4.26) $n$ represents the number of bins for each HOG.

$$BC = \sum_{i=1}^{n} \sqrt{x_i \times y_i}$$  \hspace{1cm} (4.22)

- **Chi-Square statistic (CS)**: this metric compares the synthesized and reference HOGs using (4.23).

$$CS = \sum_{i=1}^{n} \frac{(x_i - m)^2}{m}, \quad m = \frac{x_i + y_i}{2}$$  \hspace{1cm} (4.23)

- **Histogram Intersection (HI)**: this metric compares the synthesized and reference HOGs using (4.24).

$$HI = \frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} x_i}$$  \hspace{1cm} (4.24)

- **Tanimoto (TM)**: this metric compares the synthesized and reference HOGs using (4.25).

$$TM = 1 - \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} \max(x_i, y_i)}$$  \hspace{1cm} (4.25)

- **Match Distance (MD)**: this technique bases the histograms comparison on the average difference between the cumulative functions of synthesized and reference HOGs, per bin, and is computed by (4.26).
\[ MD = 1 - \frac{\sum_{i=1}^{n}|CF_x - CF_y|}{n - 1} \]  

(4.26)

Using one of the previous presented techniques for histogram comparison, a structural score is obtained per block. The final result for the assessed image is a matrix where each index is the gradient score for the correspondent block. All the algorithms of histogram comparison return a score between 0 and 1, so the score matrix is composed by values between 0 and 1.

4.5 Fusion

After obtaining the quality scores resulting from the 2D Image Quality Metric and from the Structural Quality Metric, a global quality score is achieved by fusion of these two scores. The fusion module, presented in Figure 4.1, is a delicate step because it involves the decision on how much weight should be given to each metric. Since each individual metric is computed per block, the fusion module inputs are two matrices, one per metric, where each matrix element measures quality for each block. Figure 4.14 shows an example of two images divided into four blocks - Figure 4.14 a) is the synthesized image and Figure 4.14 b) is the original image.

![Figure 4.14 Example of image division into blocks: a) Synthesized image; b) Reference image.](image)

The structure of the assessment results of the synthesized image illustrated in Figure 4.15 a) for the 2D Image Quality Metric, \( Q \), and for the Structural Quality Metric, \( S \).

![Figure 4.15 Matrices of block-based quality results for: a) 2D image quality metric (Q); b) Structural quality metric (S).](image)

The two matrices are combined in a unique matrix, \( F \), using (4.27); parameter \( \alpha \) defines how much weight is assigned to each metric.
\[ F_{ij} = a \times Q_{ij} + (1 - a) \times S_{ij} \] (4.27)

The global quality score per image, \( Q_{\text{image}} \), results from the average of the \( p\% \) lowest elements of \( F \), according to (4.28), where \( N_p \) is the number of considered elements.

\[ Q_{\text{image}} = \frac{\sum_{n=1}^{N_p} F_n}{N_p} \] (4.28)

The final result per image, which is also a score between 0 and 1, is the predicted quality of the synthesized view, as defined in Figure 4.1. Naturally, the parameters \( p \) and \( a \) have a high impact on the performance of the metric and thus, a study to find the best parameter will be presented in the next Chapter.
Chapter 5. Performance Evaluation

5.1 Introduction

In this chapter, the objective quality assessment metrics for synthesized images, proposed in Chapter 4, are evaluated. This evaluation is accomplished by comparing the objective scores resulting from the metrics, with the quality scores (e.g., MOS values) obtained from subjective tests, and for known data sets of synthesized images (e.g., IRCCyN database [59]). Since the proposed metrics are dependent on some parameters, intermediate assessments and decisions were made in order to optimize those parameters, and are also included in this chapter. The performance of the proposed metric is also evaluated relatively to known metrics, developed for conventional 2D images or for the specific case of synthesized images.

All the proposed algorithms have been implemented on Matlab R2015a, and have run on a computer with a 64-bit OS, with an Intel ® Core ™ i7-4770 CPU @ 3.40 GHz processor and 16GB RAM.

This chapter is organized as follows: section 5.2 describes the data sets of synthesized images considered in this Thesis; the logistic function used to map the metrics values to MOS values, and the statistical measurements used to validate the metrics, are presented in section 5.3; section 5.4 describes the metrics performance evaluation methodology; section 5.5 addresses the metrics parameters optimization; section 1.6 compares the proposed metrics with other solutions existing in the literature, and developed with similar purposes.

5.2 Test Material

To develop and evaluate the new metrics, two known databases of synthesized images were used: the IRCCyN database [59], provided by the Institut de Recherche en Communications et Cybernétique de Nantes, and the SIAT database [72], provided by the Shenzhen Institute of Advanced Technology. The IRCCyN database is composed by still images which were synthesized with different DIBR algorithms and using original reference images (i.e., without texture or depth maps compression). The SIAT database is composed by video sequences synthesized with the VSRS-1D-Fast software [73], using reference videos with different combinations of texture and depth maps compression levels. These two databases complement themselves, since they result in different artifacts on the synthesized images/videos.

5.2.1 IRCCyN Database

The IRCCyN database is composed by 12 original images and 84 synthesized images (96 images in total). The original images were extracted from three different MVD video sequences, with 1024x768 spatial resolution (Book Arrival, Lovebird and Newspaper), and correspond to the views (i.e., cameras) and frames detailed, respectively, on the third and second column of Table 5.1.
For the synthesis procedure, four images (referred as "Source View" in Table 5.1) were taken from each video, corresponding to the views (i.e., cameras) and frames detailed in Table 5.1 (note that these images are not available in the IRCCyN database). From each left (or right) source view, referred as "View 1" in Figure 5.1, two new images were synthesized, one corresponding to the center point of view, referred as "View 2" in Figure 5.1, and other corresponding to the right (or left) point of view, referred as "View 3" in Figure 5.1. In this figure, the red square contains the images that are provided by the IRCCyN database.
the synthesis technique. Figure 5.2 presents an image region extracted from the same synthesized view and frame, for each DIBR technique.

Table 5.2 – Considered DIBR techniques.

<table>
<thead>
<tr>
<th>Algorithm Name</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fehn Cropped [74]</td>
<td>A1</td>
</tr>
<tr>
<td>Fehn Interpolated [74]</td>
<td>A2</td>
</tr>
<tr>
<td>MPEG_VSRS [75]</td>
<td>A3</td>
</tr>
<tr>
<td>Mueller [76]</td>
<td>A4</td>
</tr>
<tr>
<td>ICME [77]</td>
<td>A5</td>
</tr>
<tr>
<td>ICIP TMM [78]</td>
<td>A6</td>
</tr>
<tr>
<td>Holes</td>
<td>A7</td>
</tr>
</tbody>
</table>

Figure 5.2 – An image region extracted from a reference view and from the synthesized views for each DIBR algorithm.

In summary, the IRCCyN database provides eight versions (seven synthesized and one original) of the same image; since there are four images per video and three different video sequences, results on a total of 96 images.

The subjective evaluation of the synthesized and original images was conducted with 43 observers and using two assessment methodologies: Absolute Category Rating with hidden Reference (ACR-HR) and Pair Comparison (PC): the ACR-HR evaluates each image individually, in a scale of "bad", "poor", "fair", "good", and "excellent", which is afterwards converted to a scale of 1 to 5 when computing the Mean Opinion Score (MOS); the PC consists in evaluating images pairs and judge which image is preferred. For the ACR-HR, the reference image is evaluated but without informing the subjects of its presence. For the PC methodology, responses from many observers yield to an interval-scale ordering of images: they can be converted to scale values using Thurstone–Mosteller’s
or Bradley–Terry's model [79] The spreadsheets with the individual scores (for both ACR-HR and PC) are provided in the IRCCyN database. Since the target of this Thesis is to create a metric to assess the absolute quality of synthesized images, the ACR-HR was used as assessment methodology, since it evaluates an image individually and not relatively to other, as in PC. The resulting MOS scores are provided in the IRCCyN database, and since the original images were also evaluated, the DMOS scores can be obtained as well.

5.2.2 SIAT-based Dataset

The SIAT database [51] contains ten different MVD sequences (Book Arrival, Ballons, Kendo, Lovebird1, Newspaper, Dancer, PoznanHall2, PoznanStreet, GT Fly and Shark), whose spatial and temporal resolutions are presented in Figure 5.3. For each MVD sequence, two source views are provided (“Input View Pair” in Figure 5.3), with 13 combinations of texture/depth compression levels (detailed in Figure 5.3) plus the uncompressed texture/depth combination; for each one of these combinations, an intermediate view was synthesized (“Output View” in Figure 5.3), resulting in 140 synthesized views. The synthesized videos can be divided into four distortion categories: four videos resulting from the UTCD (uncompressed texture and compressed depth) source views; four videos resulting from the CTD (compressed texture and uncompressed depth) source views; five videos resulting from the CTC (compressed texture and compressed depth) source views; one video resulting from the UTU (uncompressed texture and uncompressed depth) source views. All these videos (except the UTU source views), as well as the original videos corresponding to the synthesized views, are provided.

The source texture/depth view pairs were encoded with the 3DV-ATM v10.0 [80], which is the 3D-AVC based reference software for MVD coding. The algorithm used for the view synthesis was the VSRS-1D-Fast software [73].

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Resolution</th>
<th>Frame Rate</th>
<th>Synthesized Frames</th>
<th>Input View Pair</th>
<th>Output View</th>
<th>Dep. QP (U2,C2)</th>
<th>Tex. QP (C2,Y2)</th>
<th>(Tex.,Dep.) QP Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>BookArrival</td>
<td>1024 × 768</td>
<td>16.67fps</td>
<td>100</td>
<td>6 - 10</td>
<td>8</td>
<td>28,36,40,44</td>
<td>28,34,38,42</td>
<td>(22,26),(28,32),(34,36),(38,40),(42,44)</td>
</tr>
<tr>
<td>Balloons</td>
<td>1024 × 768</td>
<td>30fps</td>
<td>200</td>
<td>1 - 5</td>
<td>3</td>
<td>32,36,40,46</td>
<td>24,32,38,42</td>
<td>(24,32),(28,36),(32,40),(40,42)</td>
</tr>
<tr>
<td>Kendo</td>
<td>1024 × 768</td>
<td>30fps</td>
<td>200</td>
<td>1 - 5</td>
<td>3</td>
<td>32,38,44,48</td>
<td>24,28,32,40</td>
<td>(24,32),(32,34),(36,38),(40,44)</td>
</tr>
<tr>
<td>Lovebird1</td>
<td>1024 × 768</td>
<td>30fps</td>
<td>200</td>
<td>4 - 6</td>
<td>5</td>
<td>36,38,40,48</td>
<td>28,30,34,38</td>
<td>(28,36),(30,40),(34,44),(38,48)</td>
</tr>
<tr>
<td>Newspaper</td>
<td>1024 × 768</td>
<td>30fps</td>
<td>200</td>
<td>2 - 4</td>
<td>3</td>
<td>28,36,44,50</td>
<td>24,30,34,38</td>
<td>(28,32),(32,40),(38,44),(42,48)</td>
</tr>
<tr>
<td>Dancer</td>
<td>1920 × 1088</td>
<td>25fps</td>
<td>200</td>
<td>1 - 9</td>
<td>5</td>
<td>24,28,40,45</td>
<td>28,32,40,44</td>
<td>(24,20),(30,24),(32,28),(32,40)</td>
</tr>
<tr>
<td>PoznanHall2</td>
<td>1920 × 1088</td>
<td>25fps</td>
<td>200</td>
<td>5 - 7</td>
<td>6</td>
<td>28,32,40,46</td>
<td>24,28,32,38</td>
<td>(24,28),(26,32),(34,36),(40,42)</td>
</tr>
<tr>
<td>PoznanStreet</td>
<td>1920 × 1088</td>
<td>25fps</td>
<td>200</td>
<td>3 - 5</td>
<td>4</td>
<td>32,38,44,48</td>
<td>26,30,38,42</td>
<td>(22,28),(26,40),(34,44),(42,48)</td>
</tr>
<tr>
<td>GT Fly</td>
<td>1920 × 1088</td>
<td>25fps</td>
<td>200</td>
<td>1 - 9</td>
<td>5</td>
<td>28,36,44,48</td>
<td>24,36,40,44</td>
<td>(24,28),(32,36),(34,40),(48,44)</td>
</tr>
<tr>
<td>Shark</td>
<td>1920 × 1088</td>
<td>25fps</td>
<td>200</td>
<td>1 - 9</td>
<td>5</td>
<td>28,36,40,44</td>
<td>24,32,36,40</td>
<td>(24,28),(32,36),(36,40),(40,44)</td>
</tr>
</tbody>
</table>

*Figure 5.3 – Sequence information and texture/depth QP pairs [51].*

From the SIAT database, 60 synthesized frames and the corresponding original frames were extracted. Additionally, 120 source frame pairs, with different combinations of texture/depth compression levels, were used to synthesize a new set of images, using the View Synthesis with Inverse Mapping (VSIM) algorithm [81]. This resulted in a new data base of still images, with 180 synthesized images, the corresponding original images and the corresponding source images (see [82] for more details).
To evaluate the quality of the synthesized images, a subjective experiment was then performed [82]; this experiment was conducted in two sessions and each image was evaluated by 25 subjects in the first session, and by 18 subjects in the second session. The ACR-HR was used as assessment methodology. After the subjective assessment sessions, both MOS and DMOS values were obtained for each image under evaluation. Figure 5.4 presents the resulting MOS scores for the different images of the SIAT-based dataset.

![Figure 5.4 - Distribution of MOS scores for the SIAT-based database images.](image)

### 5.3 Logistic Function and Metrics Performance Measurement

In image quality assessment is common for people to have difficulties to distinguish the different qualities when all are of high quality, and the same happens for images with very low quality; this fact results on a subjective vs. objective scores curve having a sigmoid function behavior, as represented in Figure 5.5.

![Figure 5.5 – Logistic function representation.](image)

Taking into account this difficulty of distinguishing images with very high or very low quality, the Video Quality Expert Group (VQEG) [83] recommends the use of a logistic function regression to map the objective scores on the subjective scores, thus obtaining the predicted MOS value, MOS\(_p\). The logistic function used in this Thesis is described by (5.1).
\[ \text{MOSp} = \beta_1 + \frac{\beta_2 - \beta_1}{1 + 10^{(\beta_3-x)\beta_4}} \quad (5.1) \]

The \( \beta \) values are obtained through a regression step to minimize the error between \( \text{MOSp} \) and MOS:

- \( \beta_1 \): Is the minimum value of the curve.
- \( \beta_2 \): Is the maximum value of the curve.
- \( \beta_3 \): Is the middle value of the objective score.
- \( \beta_4 \): Is the slope of the curve.

To evaluate the objective quality assessment metrics, the following statistical measures were used (which are also recommended by VQEG):

- **Pearson Linear Correlation Coefficient (PLCC or \( r \))**: described by (5.2). In this Thesis, where \( x \) can be the objective scores or the predicted MOS values, \( \text{MOSp} \), after applying the logistic function; \( N \) is the number of assessed images.

\[
\text{PLCC} = \frac{\sum_{i=1}^{N}(\text{MOS}_i - \overline{\text{MOS}})(x_i - \overline{x})}{\sqrt{\sum_{i=1}^{N} (\text{MOS}_i - \overline{\text{MOS}})^2} \sqrt{\sum_{i=1}^{N} (x_i - \overline{x})^2}} \quad (5.2)
\]

The value of PLCC can take any value between -1 and 1, where -1 indicates a total negative linear correlation, 1 indicates a total positive linear correlation and 0 indicates that no linear correlation exists, as depicted by the scatter diagrams of Figure 5.6.

![Figure 5.6 – Scatter example and its respective PLCC.](image)

- **Spearman’s Rank Order Correlation Coefficient (SROCC)**: this coefficient evaluates how well a monotonic function [84] can represent the relation between two variables. The SROCC is described by eq. (5.3):

\[
\text{SROCC} = 1 - \frac{6}{N \times (N^2 - 1)} \sum_{i=1}^{N} (d_0(x_i, y_i))^2 \quad (5.3)
\]

with,

\[
d_0(x_i, y_i) = \text{rank}(x_i) - \text{rank}(y_i) \quad (5.4)
\]

where \( x_i \) is the MOS score and \( y_i \) is the objective score or the \( \text{MOSp} \) score for each image. The \( \text{rank}(x_i) \) and the \( \text{rank}(y_i) \) represent the positions that each value of \( x_i \) and \( y_i \) assume in their sorted list of values and \( N \) is the number of assessed images. Similarly with PLCC, the values of SROCC also range within an interval of -1 to 1.

- **Root Mean Square Error (RMSE)**: measures the amount of difference between the values of
MOS and the objective scores or MOS_p. The RMSE is given by (5.5):

\[
RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2}
\]

(5.5)

where, \( N \) is the total number of assessed images, \( x_i \) is the MOS score and \( y_i \) is the objective score or the MOS_p score.

### 5.4 Performance Assessment Methodology

To validate the quality assessment metrics proposed in Chapter 4, it is necessary to evaluate how well these metrics predict the MOS scores. Remembering the metrics’ block diagram, presented in Figure 4.1, more than one solution was proposed for each module, and each solution has parameters which need to be found: in the Structural Quality Metric module, the approach can be based on the Hausdorff distance or on the Gradient Histogram; the 2D Quality Metric module can use the MSE, the SSIM or the FSIM, as 2D metric; the Fusion module combines the results of the two previous module, and for this process the relevant parameters are \( \alpha \) and \( p \); these different possibilities are summarized in Table 5.3.

**Table 5.3 – Approaches and parameters for the different modules of the proposed metric.**

<table>
<thead>
<tr>
<th>Structural QA</th>
<th>2D Quality Metric</th>
<th>Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hausdorff based approach</td>
<td>( h(A, B) ) and ( H(A, B) ) functions</td>
<td>MSE</td>
</tr>
<tr>
<td>Gradient based approach</td>
<td>HOG voting method</td>
<td>SSIM</td>
</tr>
<tr>
<td></td>
<td>HOG comparison technique</td>
<td>FSIM</td>
</tr>
</tbody>
</table>

Of course, it is unreasonable to test all possible combinations of functions and parameters values; the methodology followed for the Hausdorff based approach was:

1. fix initial values for \( \alpha \), \( p \) (e.g., 0.5) and the 2D Quality Metric (e.g., SSIM);
2. evaluate all combinations of \( h(A, B) \), \( H(A, B) \) and normalization factors;
3. for the best normalization factor found in step 2, evaluate all combinations of \( h(A, B) \), \( H(A, B) \) and 2D Quality Metrics;
4. for the best normalization factor found in step 2, and the best combination of \( h(A, B) \), \( H(A, B) \) and 2D Quality Metric found in step 3, determine the best pair of \( \alpha \) and \( p \) values.

The methodology followed for Gradient based approach was:

1. fix initial values for \( \alpha \), \( p \) (e.g., 0.5) and disable the 2D Quality Metric (only to study which is the best voting method);
2. evaluate best voting method to obtain the HOG, and for the different histogram comparison (HC) techniques;
3. fix the HOG solution according to the result of step 2, and test the different combinations of HC techniques and 2D Quality Metrics;
4. for the best combination of HOG solution, histogram comparison technique and 2D Quality Metric, determine the best pair of \( \alpha \) and \( p \) values.

Concerning the displacement estimation algorithm and as described in Chapter 4, it is performed in three iterations, and the block size and the window search size may change at each iteration; Table 5.4 summarizes the involved parameters and their initial values, at each iteration. Note that after inspecting the IRCCyN database, the maximum displacement found between an original image and its corresponding synthesized image was around 44 pixels in the horizontal direction; on the vertical direction, no displacement was detected.

After reaching the best result for Structural Quality Metric, different block sizes and window search values were also tested for the displacement estimation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block size in horizontal and vertical directions</td>
<td>128 ( \rightarrow ) 64 ( \rightarrow ) 32 (1\textsuperscript{st}, 2\textsuperscript{nd} and 3\textsuperscript{rd} iteration)</td>
</tr>
<tr>
<td>Window search size in vertical direction (( p_1 ))</td>
<td>5 ( \rightarrow ) 3 ( \rightarrow ) 2 (1\textsuperscript{st}, 2\textsuperscript{nd} and 3\textsuperscript{rd} iteration)</td>
</tr>
<tr>
<td>Window search size in horizontal direction (( p_2 ))</td>
<td>50 ( \rightarrow ) 25 ( \rightarrow ) 13 (1\textsuperscript{st}, 2\textsuperscript{nd} and 3\textsuperscript{rd} iteration)</td>
</tr>
</tbody>
</table>

After reaching the best result for Structural Quality Metric, different block sizes and window search values were also tested for the displacement estimation.

### 5.5 Results and Analysis

This section presents and discusses the results of the quality metrics assessment experiments.

#### 5.5.1 Edge Detection

As mentioned in Chapter 4, objects edges are detected using the Canny edge detection algorithm. In this step, the requirement for finding the true image edges is of upmost importance, which means excluding the false edges resulting from image noise. In the Canny detector, and after computing the image gradient, a potential edge pixel is classified as edge or non-edge based on a comparison with a threshold. To find out the best threshold selection method for the Canny edge detection, some assessment study is necessary.

Actually, in the used Matlab function implementing the Canny detector, the threshold is a two element vector, where \( T \) is the upper threshold and \( 0.4 \times T \) is the lower threshold; If \( T \) is not specified, the function sets automatically the thresholds. Experimentally, it was verified that when \( T \) is automatically set, the edge detection results in too many edges, which may jeopardize the results.

Figure 5.7 illustrates the impact of varying the value of the threshold, \( T \), on the detected edges of an original image represented in Figure 5.7 a); for the edges detection, only the luminance component was used. Since the goal is to minimize the presence of false edges, the best result was obtained for \( T = 0.1 \), corresponding to Figure 5.7 e). This threshold filtrates most of the noisy edges present in
Figure 5.7 f), where $T = 0.05$, and also in Figure 5.7 b), with automatic thresholds, but without losing important information. In Figure 5.7 c) and d), where $T = 0.2$ and $T = 0.15$, respectively, some important edges were lost (e.g., in the building front). Thus, the threshold for the Canny edge detection was set to $T = 0.1$. The edge images for the other image sequences of IRCCyN and SIAT-based datasets are available in Appendix A, using $T = 0.1$. and automatic $T$.

![Figure 5.7 – Canny edge detection: a) Original image; b) Edge image using automatic $T$; c) Edge image using $T = 0.2$; d) Edge image using $T = 0.15$; e) Edge image using $T = 0.1$; f) Edge image using $T = 0.05$.](image)

### 5.5.2 Hausdorff Distance Assessment Results for the IRCCyN Database

Let $A$ and $B$ represent, respectively, a shift compensated synthesized image and its corresponding original image. As described in the Chapter 4, the first step in the Hausdorff distance based procedure is to compute for each edge pixel in $A$, the nearest edge pixel in $B$, and for each edge pixel $B$, the nearest edge pixel in $A$; this results, respectively, in matrix $d(a,B)$ and in matrix $d(b,A)$, where each matrix element is the distance of an edge to its nearest edge in the other image. Each matrix is then split in $N \times N$ pixel blocks, and the Hausdorff distance is computed for each block, applying the $h(.)$ and $H(.)$ functions, described in Chapter 4.

The $\alpha$ and $p$ parameters were both initially settled to 0.5; $N$ was fixed in 32, since it corresponds to the lowest block size considered on the displacement estimation algorithm.

The first considered metric combines the Hausdorff distance with SSIM. Table 5.5 presents the resulting PLCC values between the MOS scores and the values resulting from the metric, for the different $h(.)$ and $H(.)$ functions combinations. As in [56], the normalization factor (NF) used on $h_1$ to $h_6$ was the block size, $N=32$. For the functions $h_7$, $h_8$ and $h_9$, and since these functions do not correspond to a distance (as the previous ones), the normalization factor needed to be studied more thoroughly. Hence, based on the histograms of Figure 4.10 and Figure 4.11, different normalization factors were tested to find out which one maximizes the PLCC between the metric values and the MOS scores; the selected values are presented in the NF column of Table 5.5. These NF values
correspond to the highest values of the $h(.)$ function, found in the histograms of Figure 4.10 and Figure 4.11.

The value of the $\delta$ parameter required in $h_8$ and $h_9$ ((4.8) and (4.9)), was initially settled to 5 as, by inspection of the synthesized images, it corresponds to the limit value above which the edges distortion becomes visible.

Table 5.5 - PLCC between the MOS scores and Hausdorff distance combined with SSIM.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$\alpha$</th>
<th>$H_2$</th>
<th>$H_3$</th>
<th>$H_4$</th>
<th>NF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_2$</td>
<td>0.6308</td>
<td>0.5979</td>
<td>0.6244</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>$h_3$</td>
<td>0.6662</td>
<td>0.6269</td>
<td>0.6529</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>$h_4$</td>
<td>0.6969</td>
<td>0.6597</td>
<td>0.6829</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>$h_5$</td>
<td>0.7085</td>
<td>0.6750</td>
<td>0.6977</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>$h_6$</td>
<td>0.6408</td>
<td>0.6062</td>
<td>0.6324</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>$h_7$</td>
<td>0.6881</td>
<td>0.6957</td>
<td>0.6969</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>$h_7$</td>
<td>0.7173</td>
<td>0.7096</td>
<td>0.7140</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>$h_7$</td>
<td>0.7253</td>
<td>0.7081</td>
<td>0.7160</td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>$h_7$</td>
<td>0.6986</td>
<td>0.6618</td>
<td>0.6786</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>$h_8$</td>
<td>0.7372</td>
<td>0.7445</td>
<td>0.7447</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>$h_8$</td>
<td>0.7433</td>
<td>0.7413</td>
<td>0.7429</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>$h_8$</td>
<td>0.7455</td>
<td>0.7336</td>
<td>0.7371</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>$h_8$</td>
<td>0.7384</td>
<td>0.7094</td>
<td>0.7193</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>$h_9$</td>
<td>0.7342</td>
<td>0.7464</td>
<td>0.7455</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>$h_9$</td>
<td>0.7466</td>
<td>0.7456</td>
<td>0.7459</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>$h_9$</td>
<td>0.7447</td>
<td>0.7243</td>
<td>0.7289</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>$h_9$</td>
<td>0.7336</td>
<td>0.7033</td>
<td>0.7121</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>$h_9$</td>
<td>0.7214</td>
<td>0.6849</td>
<td>0.6976</td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>$h_9$</td>
<td>0.6699</td>
<td>0.6279</td>
<td>0.6461</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

Based on the results of Table 5.5, the normalization factors for $h_7$, $h_8$ and $h_9$ should be chosen around 400, 15 and 100, respectively (signalized in green); these values were kept for the rest of the experiments.

Using the previously obtained normalization factors, the correlation results were recalculated using different 2D Quality Metrics. Although the SSIM and FSIM return values in an interval from 0 to 1, the MSE does not. To normalize the MSE values, (5.6) was used, where $\max_{MSE}$ and $\min_{MSE}$ are, respectively, the maximum and minimum MSE values for the whole set of images in the IRCCyN database.

$$MSE_{normalized} = \frac{MSE - \min_{MSE}}{\max_{MSE} - \min_{MSE}} \quad (5.6)$$

Table 5.6 to Table 5.8 present the PLCC results using SSIM, MSE and FSIM, respectively, and for the different combinations of $h(.)$ and $H(.)$ functions; for each case, the three best results are signalized in green.
These results show that the best Hausdorff distances are the ones relying on the edge pixel distances superior to \( \delta \), namely \( h_8 \) and \( h_9 \); in fact, \( \delta \) forces the technique to consider the most significant edges distortions that may effectively affect the users’ perceived quality, instead of including the small, but numerous, edges differences that may occur in high textured areas, but which have less impact on the users.

According to the results presented in the previous tables, the best results were achieved for \((h_8, H_2)\), \((h_8, H_3)\) and \((h_9, H_4)\) and when using SSIM or FSIM in the 2D Quality Metric module; other \((h,H)\) combinations, as well as the MSE metric, were then disregarded for the following assessments.

The next two tables present the PLCC values between MOS scores and metric values, for the three best \((h,H)\) combinations, and varying \( \alpha \); the considered 2D quality metrics were SSIM (Table 5.9), and FSIM (Table 5.10). It is worth to note that in both cases there is a large range of \( \alpha \) values with quite similar PLCC values; the best results are signalized in green.
Table 5.9 – Impact of α for the three best (h, H) combinations using SSIM as 2D Quality Metric.

<table>
<thead>
<tr>
<th>p</th>
<th>α</th>
<th>(h_9, H_2)</th>
<th>(h_9, H_3)</th>
<th>(h_9, H_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
<td>0.4782</td>
<td>0.4782</td>
<td>0.4782</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.5758</td>
<td>0.5629</td>
<td>0.5654</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.6647</td>
<td>0.6474</td>
<td>0.6508</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>0.7154</td>
<td>0.7020</td>
<td>0.7045</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.7383</td>
<td>0.7316</td>
<td>0.7329</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.7466</td>
<td>0.7456</td>
<td>0.7459</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td><strong>0.7477</strong></td>
<td>0.7507</td>
<td>0.7505</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.7458</td>
<td><strong>0.7511</strong></td>
<td><strong>0.7506</strong></td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.7427</td>
<td>0.7492</td>
<td>0.7487</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.7391</td>
<td>0.7464</td>
<td>0.7459</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.7355</td>
<td>0.7432</td>
<td>0.7429</td>
</tr>
</tbody>
</table>

Table 5.10 - Impact of α for the three best (h, H) combinations using FSIM as 2D Quality Metric.

<table>
<thead>
<tr>
<th>p</th>
<th>α</th>
<th>(h_9, H_2)</th>
<th>(h_9, H_3)</th>
<th>(h_9, H_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
<td>0.4883</td>
<td>0.4883</td>
<td>0.4883</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.6467</td>
<td>0.6287</td>
<td>0.6319</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.7237</td>
<td>0.7098</td>
<td>0.7122</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>0.7441</td>
<td>0.7386</td>
<td>0.7395</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td><strong>0.7478</strong></td>
<td>0.7478</td>
<td>0.7479</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.7467</td>
<td><strong>0.7500</strong></td>
<td><strong>0.7497</strong></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>0.7445</td>
<td>0.7494</td>
<td>0.7490</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.7420</td>
<td>0.7480</td>
<td>0.7476</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.7396</td>
<td>0.7464</td>
<td>0.7460</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.7374</td>
<td>0.7448</td>
<td>0.7444</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.7355</td>
<td>0.7432</td>
<td>0.7429</td>
</tr>
</tbody>
</table>

After obtaining the α that maximizes the PLCC value - 0.3 and 0.4 for SSIM and 0.5 and 0.6 for FSIM - the same procedure was repeated to find the best value for p. Results are presented in Table 5.11 and Table 5.12. As for the α parameter, there is also a large range of p values, namely [0.4,1.0], with similar PLCC values; the best results are signalized in green.

Table 5.11 - Impact of p for the three best (h, H) combinations using the SSIM as 2D metric.

<table>
<thead>
<tr>
<th>α</th>
<th>p</th>
<th>(h_9, H_2)</th>
<th>(h_9, H_3)</th>
<th>(h_9, H_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>1</td>
<td>0.7467</td>
<td>0.7486</td>
<td>0.7484</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.7467</td>
<td>0.7487</td>
<td>0.7484</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.7465</td>
<td>0.7485</td>
<td>0.7483</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>0.7468</td>
<td>0.7488</td>
<td>0.7485</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.7474</td>
<td>0.7495</td>
<td>0.7491</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td><strong>0.7477</strong></td>
<td>0.7511</td>
<td>0.7506</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4</td>
<td>0.7418</td>
<td><strong>0.7541</strong></td>
<td><strong>0.7531</strong></td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.7055</td>
<td>0.7453</td>
<td>0.7381</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.6284</td>
<td>0.6794</td>
<td>0.6609</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.5580</td>
<td>0.5532</td>
<td>0.5481</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
</tbody>
</table>
Table 5.12 - Impact of p for the three best (h, H) combinations using the FSIM as 2D metric.

<table>
<thead>
<tr>
<th>α</th>
<th>p</th>
<th>(h₉, H₂)</th>
<th>(h₉, H₃)</th>
<th>(h₉, H₄)</th>
</tr>
</thead>
<tbody>
<tr>
<td>α=0.6 for (h₉, H₂)</td>
<td>1</td>
<td>0.7444</td>
<td>0.7463</td>
<td>0.7462</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.7446</td>
<td>0.7466</td>
<td>0.7464</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.7449</td>
<td>0.7467</td>
<td>0.7466</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>0.7455</td>
<td>0.7471</td>
<td>0.7470</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.7462</td>
<td>0.7479</td>
<td>0.7477</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.7478</td>
<td>0.7500</td>
<td>0.7497</td>
</tr>
<tr>
<td>α=0.5 for (h₉, H₃) and (h₉, H₄)</td>
<td>0.4</td>
<td>0.7428</td>
<td>0.7533</td>
<td>0.7525</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.7061</td>
<td>0.7435</td>
<td>0.7366</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.6226</td>
<td>0.6739</td>
<td>0.6566</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.5213</td>
<td>0.5380</td>
<td>0.5303</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
</tbody>
</table>

At this point, the α and p values that maximize the PLCC between the proposed metric and the MOS scores were found, which are summarized in Table 5.13.

Table 5.13 – Parameters that maximize PLCC between the metric values and MOS scores.

<table>
<thead>
<tr>
<th>p</th>
<th>α</th>
<th>2D Quality Metric</th>
<th>(h₉, H₂)</th>
<th>(h₉, H₃)</th>
<th>(h₉, H₄)</th>
</tr>
</thead>
<tbody>
<tr>
<td>p=0.5 for (h₀, H₂)</td>
<td>α=0.4 for (h₀, H₂)</td>
<td>SSIM</td>
<td>0.7477</td>
<td>0.7541</td>
<td>0.7531</td>
</tr>
<tr>
<td>p=0.4 for (h₀, H₃)</td>
<td>α=0.3 for (h₀, H₃)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p=0.4 for (h₀, H₄)</td>
<td>α=0.3 for (h₀, H₄)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>α=0.6 for (h₀, H₂)</td>
<td>FSIM</td>
<td>0.7478</td>
<td>0.7533</td>
<td>0.7525</td>
</tr>
<tr>
<td></td>
<td>α=0.5 for (h₀, H₃)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>α=0.5 for (h₀, H₄)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bearing in mind that equation h₀ depends on δ, the effect of varying this value was assessed using (h₀, H₃) combined with SSIM. Figure 5.8 shows the evolution of PLCC for different values of δ, allowing to conclude that the δ which maximizes the correlation is around 5.5 (PLCC=0.7548).

Figure 5.8 – Variations of PLCC of (h₀, H₃)+SSIM with MOS for different δ.

According to Figure 5.8, for δ<5.5, the lower is δ, the lower is the correlation - this happens because if all the small distances are considered and added up, the estimated distortion will have an high value but not reflecting the MOS score, since those small distances have little impact on
perceived quality. The same happens if only the highest distances are considered - in this case, some distortions that may affect the users’ perception of quality will be ignored.

The PLCC was recalculated for the three best combinations of \( h(\cdot) \) and \( H(\cdot) \), with \( \delta = 5.5 \), and with SSIM and FSIM as 2D quality metrics. After that, the logistic function described previously was applied to fit the objective scores to the subjective MOS scores. Table 5.14 presents the PLCC results before and after applying the logistic function. The highest PLCC was achieved for \((h_9, H_3)\) combined with SSIM, with \( \alpha = 0.3 \) and \( p = 0.4 \).

<table>
<thead>
<tr>
<th>( p )</th>
<th>( \alpha )</th>
<th>2D Metric</th>
<th>Logistic</th>
<th>( (h_9, H_2) )</th>
<th>( (h_9, H_3) )</th>
<th>( (h_9, H_4) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5 for ((h_9, H_2))</td>
<td>( \alpha = 0.4 ) for ((h_9, H_2))</td>
<td>PLCC</td>
<td>no</td>
<td>0.7515</td>
<td>0.7548</td>
<td>0.7546</td>
</tr>
<tr>
<td></td>
<td>( \alpha = 0.3 ) for ((h_9, H_3))</td>
<td>SROCC</td>
<td>no</td>
<td>0.7220</td>
<td>0.7158</td>
<td>0.7148</td>
</tr>
<tr>
<td>0.4 for ((h_9, H_3))</td>
<td>( \alpha = 0.3 ) for ((h_9, H_4))</td>
<td>RMSE</td>
<td>no</td>
<td>1.5356</td>
<td>1.5600</td>
<td>1.5760</td>
</tr>
<tr>
<td></td>
<td>( \alpha = 0.6 ) for ((h_9, H_2))</td>
<td>PLCC</td>
<td>yes</td>
<td>0.7507</td>
<td>0.7533</td>
<td>0.7534</td>
</tr>
<tr>
<td></td>
<td>( \alpha = 0.5 ) for ((h_9, H_3))</td>
<td>SROCC</td>
<td>yes</td>
<td>0.7597</td>
<td>0.7621</td>
<td>0.7614</td>
</tr>
<tr>
<td>0.4 for ((h_9, H_4))</td>
<td>( \alpha = 0.5 ) for ((h_9, H_4))</td>
<td>RMSE</td>
<td>no</td>
<td>1.4389</td>
<td>1.4707</td>
<td>1.4819</td>
</tr>
<tr>
<td></td>
<td>( \alpha = 0.5 ) for ((h_9, H_4))</td>
<td>FSIM</td>
<td>yes</td>
<td>0.4122</td>
<td>0.4104</td>
<td>0.4109</td>
</tr>
</tbody>
</table>

Figure 5.9 a) presents the scatter plot of the objective scores of the optimized metric (Haudorff distance with \( \delta = 5.5 \), SSIM as 2D metric, \( \alpha = 0.3 \) and \( p = 0.4 \)) versus the MOS scores. The red curve represents the resulting logistic function; the values for the set of \( \beta \) parameters, obtained through nonlinear regression, were: \( \beta_1 = 0.9768 \), \( \beta_2 = 3.0587 \), \( \beta_3 = 0.5873 \) and \( \beta_4 = 3.9227 \). Figure 5.9 b) presents the scatter plot of the fitted metric values (MOSp) versus MOS scores.

![Figure 5.9 a)](image1)

![Figure 5.9 b)](image2)

**Figure 5.9** – Optimized metric values vs MOS scores: a) before the logistic fitting; b) after the fitting.

Figure 5.9 b) shows that the proposed metric correlates quite well with the MOS scores for the images with the lowest quality, but not so well for the highest quality images. To understand the
reason behind this, the set of images with MOS$_p$=3 were analyzed. One thing in common among these images is that almost all were synthesized from the same reference image, shown in Figure 5.10 a). To find out why there is such a difference in MOS for these images, the image with the worst MOS from this set was analyzed more deeply (see Figure 5.10 b)); the reference image was evaluated with a MOS score of 4.28 and the synthesized one with a MOS of 2.21.

After inspecting the images, it is possible to notice that there are some distortions around the face of the men, and on the clock behind him, but which are indeed quite small to be detected by an automatic tool. Although the image seems to have a good global quality and the existing distortions do not seem very outstanding, the fact that they are in, and around, an human face probably caught the users’ attention and had higher impact in their perception of quality.

![Figure 5.10 – a) Reference image; b) Synthesized image.](image)

5.5.3 Gradient Based Metric Assessment Results for the IRCCyN Database

Similarly to the Hausdorff distance based approach, the gradient is also computed for the whole synthesized and reference images, using the luminance component; the gradient of the synthesized image is then compensated for eventual consistent shifts. Both gradient images - the one corresponding to the original view and the one corresponding to the synthesized view and already compensated for shifts - are divided in blocks, and the HOG is generated per block. In the HOG computation, and to reduce the influence of noise, only gradient magnitudes above 75 were considered; this threshold was obtained by inspecting the gradient magnitude images. The histogram comparison (HC) techniques are then applied at the block level.

As explained in the previous chapter, two voting solutions were considered to compute the HOG. In order to find out the best approach, the PLCC between the MOS scores and the structural metric score - resulting from the two HOG approaches and the different HC methods - were computed and are presented Table 5.15 in and Table 5.16.
From these results, the decision was to choose the HOG which uses the pixel gradient magnitude as vote weight since it has better performance for all histogram comparison techniques. The HoG with pixel gradient magnitude was used in the following assessments.

Repeating the methodology that was adopted for the Hausdorff distance assessment, the $\alpha$ and $p$ values were both set to 0.5. Then, for the different combinations of HC technique and 2D Quality Metric (SSIM, FSIM and MSE), the PLCC between the MOS values and the objective scores was computed and is presented in Table 5.17.

Based on the results of Table 5.17, both SSIM and FSIM were maintained in the following test. Table 5.18 and Table 5.19 show the resulting PLCC for SSIM and FSIM, respectively, plus HOG, using the best HC techniques and varying the value of $\alpha$.

---

Table 5.15 – PLCC using the HOG with the gradient magnitude as vote weight.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$\alpha$</th>
<th>Histogram Comparison Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0</td>
<td>E</td>
</tr>
</tbody>
</table>

Table 5.16 - PLCC using the HOG with the number of pixels as vote weight.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$\alpha$</th>
<th>Histogram Comparison Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0</td>
<td>E</td>
</tr>
</tbody>
</table>

Table 5.17 – PLCC for different combinations of 2D Metric and HC technique.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$p$</th>
<th>2D Metric</th>
<th>Histogram Comparison Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>SSIM</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FSIM</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSE</td>
<td>E</td>
</tr>
</tbody>
</table>

Based on the results of Table 5.17, both SSIM and FSIM were maintained in the following test. Table 5.18 and Table 5.19 show the resulting PLCC for SSIM and FSIM, respectively, plus HOG, using the best HC techniques and varying the value of $\alpha$. 

---
Table 5.18 - Variation of α for the histogram comparison techniques using SSIM as 2D metric.

<table>
<thead>
<tr>
<th>p</th>
<th>α</th>
<th>Histogram Comparison Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>0.4782</td>
<td>0.4782</td>
</tr>
<tr>
<td>0.9</td>
<td>0.5131</td>
<td>0.4973</td>
</tr>
<tr>
<td>0.8</td>
<td>0.5533</td>
<td>0.5257</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5906</td>
<td>0.5551</td>
</tr>
<tr>
<td>0.6</td>
<td>0.6211</td>
<td>0.5830</td>
</tr>
<tr>
<td>0.5</td>
<td>0.6419</td>
<td>0.6074</td>
</tr>
<tr>
<td>0.4</td>
<td><strong>0.6517</strong></td>
<td>0.6274</td>
</tr>
<tr>
<td>0.3</td>
<td>0.6511</td>
<td>0.6425</td>
</tr>
<tr>
<td>0.2</td>
<td>0.6426</td>
<td>0.6524</td>
</tr>
<tr>
<td>0.1</td>
<td>0.6292</td>
<td>0.6571</td>
</tr>
<tr>
<td>0</td>
<td>0.6146</td>
<td><strong>0.6575</strong></td>
</tr>
</tbody>
</table>

Table 5.19 - Variation of α for the histogram comparison techniques using FSIM as 2D metric.

<table>
<thead>
<tr>
<th>p</th>
<th>α</th>
<th>Histogram Comparison Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>0.4883</td>
<td>0.4883</td>
</tr>
<tr>
<td>0.9</td>
<td>0.5545</td>
<td>0.5212</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6109</td>
<td>0.5631</td>
</tr>
<tr>
<td>0.7</td>
<td>0.6440</td>
<td>0.5957</td>
</tr>
<tr>
<td>0.6</td>
<td>0.6576</td>
<td>0.6192</td>
</tr>
<tr>
<td>0.5</td>
<td><strong>0.6585</strong></td>
<td>0.6354</td>
</tr>
<tr>
<td>0.4</td>
<td>0.6523</td>
<td>0.6460</td>
</tr>
<tr>
<td>0.3</td>
<td>0.6430</td>
<td>0.6525</td>
</tr>
<tr>
<td>0.2</td>
<td>0.6328</td>
<td>0.6561</td>
</tr>
<tr>
<td>0.1</td>
<td>0.6230</td>
<td>0.6575</td>
</tr>
<tr>
<td>0</td>
<td>0.6146</td>
<td><strong>0.6575</strong></td>
</tr>
</tbody>
</table>

Following the same procedure as for the Hausdorff distance assessment, α was fixed on the value that maximizes the correlation. Then, different values of p were tested - the resulting PLCC values are represented in Table 5.20 using SSIM as 2D quality metric and in Table 5.21 using FSIM.
Table 5.20 - PLCC for HOG+SSIM, and varying p.

<table>
<thead>
<tr>
<th>α</th>
<th>p</th>
<th>Histogram Comparison Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>E</td>
</tr>
<tr>
<td>α=0.4 for E</td>
<td>1</td>
<td>0.6451</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.6469</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.6477</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>0.6491</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.6508</td>
</tr>
<tr>
<td>α=0 for CS</td>
<td>0.5</td>
<td>0.6517</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>0.6516</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.6523</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.6526</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.6372</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>NaN</td>
</tr>
</tbody>
</table>

Table 5.21 - PLCC for HOG+FSIM, and varying p.

<table>
<thead>
<tr>
<th>α</th>
<th>p</th>
<th>Histogram Comparison Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>E</td>
</tr>
<tr>
<td>α=0.5 for E</td>
<td>1</td>
<td>0.6510</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.6518</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.6530</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>0.6545</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.6562</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.6585</td>
</tr>
<tr>
<td>α=0 for CS</td>
<td>0.4</td>
<td>0.6612</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.6652</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.6693</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.6550</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>NaN</td>
</tr>
</tbody>
</table>

The logistic function was the applied for the best combination of metrics and parameters values, in order to fit the objective values to the subjective scores. Table 5.22 shows the PLCC, Spearman and RMSE results after applying the fitting.
Table 5.22 - Best achieved correlation results for the HOG based approach.

<table>
<thead>
<tr>
<th>p</th>
<th>α</th>
<th>2D Metric</th>
<th>Logistic</th>
<th>Histogram Comparison Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E</td>
</tr>
<tr>
<td>E: α=0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS: α=0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HI: α=0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TM: α=0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>α=0.4 for E</td>
<td>PLCC</td>
<td>no</td>
<td>0.6526</td>
<td>0.6575</td>
</tr>
<tr>
<td>α=0 for CS</td>
<td></td>
<td>yes</td>
<td>0.6633</td>
<td>0.664</td>
</tr>
<tr>
<td>α=0 for HI</td>
<td>SROCC</td>
<td>no</td>
<td>0.6696</td>
<td>0.6453</td>
</tr>
<tr>
<td>α=0.2 TM</td>
<td></td>
<td>yes</td>
<td>0.6696</td>
<td>0.6453</td>
</tr>
<tr>
<td>α=0.5 for E</td>
<td>RMSE</td>
<td>no</td>
<td>1.5117</td>
<td>1.3995</td>
</tr>
<tr>
<td>α=0 for CS</td>
<td></td>
<td>yes</td>
<td>0.4744</td>
<td>0.4740</td>
</tr>
<tr>
<td>α=0 for HI</td>
<td>PLCC</td>
<td>no</td>
<td>0.6693</td>
<td>0.6575</td>
</tr>
<tr>
<td>α=0.2 TM</td>
<td></td>
<td>yes</td>
<td>0.6841</td>
<td>0.6640</td>
</tr>
<tr>
<td>α=0.5 for E</td>
<td>SROCC</td>
<td>no</td>
<td>0.6895</td>
<td>0.6453</td>
</tr>
<tr>
<td>α=0 for CS</td>
<td></td>
<td>yes</td>
<td>0.6895</td>
<td>0.6453</td>
</tr>
<tr>
<td>α=0 for HI</td>
<td>RMSE</td>
<td>no</td>
<td>1.4464</td>
<td>1.3995</td>
</tr>
<tr>
<td>α=0.3 TM</td>
<td></td>
<td>yes</td>
<td>0.4624</td>
<td>0.4740</td>
</tr>
</tbody>
</table>

The obtained results show that Histogram Intersection (HI) technique reaches the highest PLCC among the different HC techniques; the values for the set of β parameters, required by the logistic function, obtained through nonlinear regression, are: β₁=0.6203, β₂=3.224, β₃=0.7194 and β₄=5.135.

Figure 5.11 a) presents the scatter plot of the objective values of the optimized metric versus MOS scores for the HI metric; Figure 5.11 b) presents the scatter plot of the fitted metric values (MOSp) versus MOS scores. By inspecting Figure 5.11 a) it is possible to see that for the same MOS score of 1, the objective scores range from 0.5 to almost 0.85, which did not happen for the Hausdorff based approach. The synthesized images that have MOS scores of 1 are those corresponding to the A7 DIBR technique, for which there is no inpainting on the resulting holes, whose impact is underestimated by the gradient based approach.

![Figure 5.11 - Objective Score vs MOS for Gradient based approach before and after the logistic fitting: a) and b.](image-url)
5.5.4 Displacement Estimation Assessment

The displacement estimation is responsible for guaranteeing that the synthesized image is spatially synchronized with its correspondent original image. For this operation to be effective, choosing an appropriate window search is rather important. Knowing that the highest displacement in the IRCCyN database, between an original and a synthesized image, is around 44 pixels in the horizontal direction, the window search should not be smaller than this value. Table 5.23 shows the PLCC for the Hausdorff distance based approach plus SSIM, using different window search sizes. Table 5.24 shows the block size impact.

The obtained results showed that the best window search is the one who approximates the size of the displacement and is halved in each iteration, namely 50-25-13. Using a window search size higher than the displacement can lead to wrong shifts estimations while using a smaller window search can lead to bypass some object shifts. In Table 5.24, the results show that from iteration to iteration the PLCC gets higher, concluding that the use of iterations is a good approach to achieve better shifts estimations and compensations.

Table 5.23 – Impact on PLCC using different window search sizes.

<table>
<thead>
<tr>
<th>Structural Metric</th>
<th>2D Metric</th>
<th>Logistic</th>
<th>Window search size (1st-2nd-3rd iteration)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>70-35-18</td>
</tr>
<tr>
<td>Hausdorff distance</td>
<td>SSIM</td>
<td>no</td>
<td>0.7432</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>0.7480</td>
</tr>
</tbody>
</table>

Table 5.24 – Impact on PLCC using an hierarchical displacement estimation technique.

<table>
<thead>
<tr>
<th>Structural Metric</th>
<th>2D Metric</th>
<th>Logistic</th>
<th>Block size (1st-2nd-3rd iteration)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>128-64-32</td>
</tr>
<tr>
<td>Hausdorff distance</td>
<td>SSIM</td>
<td>no</td>
<td>0.6926</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>0.6959</td>
</tr>
</tbody>
</table>

5.5.5 Final Results

Table 5.25 presents the final results for the proposed metrics and its relevant parameters.

Table 5.25 – Final Results for the proposed metric.

<table>
<thead>
<tr>
<th>Structural Quality Metric</th>
<th>2D Quality Metric</th>
<th>Block Size</th>
<th>Window Search</th>
<th>α</th>
<th>p</th>
<th>δ</th>
<th>PLCC</th>
<th>SROCC</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hausdorff distance with</td>
<td>SSIM</td>
<td>128-64-32</td>
<td>50-25-13</td>
<td>0.3</td>
<td>0.4</td>
<td>5.5</td>
<td><strong>0.7643</strong></td>
<td>0.7158</td>
<td>0.4088</td>
</tr>
<tr>
<td>(h₀,H₃)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gradient based (using HI)</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0.6</td>
<td>-</td>
<td><strong>0.6852</strong></td>
<td>0.6513</td>
<td>0.4617</td>
</tr>
</tbody>
</table>

After inspecting the results presented in Table 5.25 the Hausdorff distance was the approach kept for the Structural Quality Metric for the metrics comparison, presented in the next section.
5.6 Final Evaluation of the Proposed Metrics

In this section, the proposed metric was compared with conventional quality metrics developed for 2D images and video but also with quality metrics designed specifically for synthesized images. These comparisons were conducted in two phases that use, respectively, the IRCCyN database and the SIAT-based dataset.

5.6.1 Metrics Comparison Using the IRCCyN Database

Table 5.26 presents the PLCC between MOS scores and the predicted MOS values, for conventional 2D quality metrics and for the proposed one; all the metrics use the same logistic function, expressed in eq. (5.1); the improvement brought by the proposed metric is evident since these metrics typically fail for the type of degradations caused by the image based rendering approach. For the 2D quality metrics no shifts compensation was applied, the metrics were used in their traditional way.

Table 5.27 presents the resulting PLCC for each metric. For the DSQM and 3DSwIM metrics, the objective scores were obtained using the source code provided by the authors; the VCIP2013 metric was implemented in this Thesis; the SSIM-based VSQA results were picked-up from [55]. Apart of the proposed metric, which used MOS score as subjective score, the remaining metrics used DMOS. For the DSQM metric, the considered reference view was the original one, while on [66] the source views was used as reference, resulting in a much higher PLCC value (PLCC=0.7588). The remaining conditions are the same as described in the papers. Also in this case, the proposed metric has the best performance.

Table 5.26 – Correlation results between MOS and objective values for conventional 2D Quality Metrics using the IRCCyN database.

<table>
<thead>
<tr>
<th>PSNR</th>
<th>SSIM</th>
<th>VSNR</th>
<th>MSSIM</th>
<th>FSIM</th>
<th>MSE</th>
<th>NQM</th>
<th>WSNR</th>
<th>VIFP</th>
<th>VIF</th>
<th>Prop</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLCC</td>
<td>0.5415</td>
<td>0.4888</td>
<td>0.4071</td>
<td>0.3249</td>
<td>0.5294</td>
<td>0.5455</td>
<td>0.5338</td>
<td>0.5598</td>
<td>0.3253</td>
<td>0.4235</td>
</tr>
<tr>
<td>SROCC</td>
<td>0.4835</td>
<td>0.3352</td>
<td>0.1442</td>
<td>0.2246</td>
<td>0.4696</td>
<td>0.4829</td>
<td>0.4585</td>
<td>0.4748</td>
<td>0.2478</td>
<td>0.2323</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.5329</td>
<td>0.5530</td>
<td>0.5790</td>
<td>0.6030</td>
<td>0.5378</td>
<td>0.5313</td>
<td>0.5360</td>
<td>0.5252</td>
<td>0.5994</td>
<td>0.5742</td>
</tr>
</tbody>
</table>

Table 5.27 – Correlation results between the subjective and objective quality scores for IRCCyN database.

<table>
<thead>
<tr>
<th>DSQM</th>
<th>3DSwIM</th>
<th>SSIM-based VSQA</th>
<th>VCIP2013</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLCC</td>
<td>0.6192</td>
<td>0.6864</td>
<td>0.6142</td>
<td>0.6447</td>
</tr>
<tr>
<td>SROCC</td>
<td>0.4106</td>
<td>0.6156</td>
<td>-</td>
<td>0.5989</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.5228</td>
<td>0.4842</td>
<td>-</td>
<td>0.5090</td>
</tr>
</tbody>
</table>
Figure 5.12 presents the scatter plot of the fitted metric values (MOSp or DMOSp) versus the subjective scores (MOS or DMOS) for each technique presented in Table 5.27, except SSIM-based VSQA, which the scores could not be reproduced.

![Figure 5.12](image)

Figure 5.12 – Scatter plot of fitted objective scores vs subjective scores referring to the IRCCyN database for the following metrics: a) DSQM; b) 3DSwIM; c) VCIP2013; d) Proposed.

### 5.6.2 Metrics Comparison Using the SIAT-based Dataset

To assess how the proposed metric would perform in other datasets, the SIAT-based database was used. As described previously, this database contains synthesized images whose source views were also compressed. Table 5.28 shows the PLCC obtained for conventional 2D quality metrics. It is worth to note that some of this metrics (namely, SSIM and MSSIM) performs much better on this database than on the IRCCyN database; this may be justified by the existence of compression artifacts on the synthesized images (inherit from the source images) - for which the conventional 2D quality metrics are well adapted - and that in some cases mask the synthesis artifacts reducing its perceptual impact.
Table 5.28 – Correlation results between MOS and objective score for traditional 2D Quality Metrics for SIAT-based database.

<table>
<thead>
<tr>
<th>Metric</th>
<th>PLCC</th>
<th>SROCC</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>0.6860</td>
<td>0.6733</td>
<td>0.6517</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.7496</td>
<td>0.7565</td>
<td>0.5928</td>
</tr>
<tr>
<td>VSNR</td>
<td>0.3998</td>
<td>0.3300</td>
<td>0.8210</td>
</tr>
<tr>
<td>MSSIM</td>
<td>0.8160</td>
<td>0.8003</td>
<td>0.5178</td>
</tr>
<tr>
<td>FSIM</td>
<td>0.6812</td>
<td>0.6917</td>
<td>0.6557</td>
</tr>
<tr>
<td>MSE</td>
<td>0.6853</td>
<td>0.6733</td>
<td>0.6523</td>
</tr>
<tr>
<td>NQM</td>
<td>0.3451</td>
<td>0.2760</td>
<td>0.8409</td>
</tr>
<tr>
<td>WSNR</td>
<td>0.5252</td>
<td>0.4901</td>
<td>0.7622</td>
</tr>
<tr>
<td>VIFP</td>
<td>0.7362</td>
<td>0.7383</td>
<td>0.6062</td>
</tr>
<tr>
<td>VIF</td>
<td>0.6773</td>
<td>0.6541</td>
<td>0.6590</td>
</tr>
</tbody>
</table>

The resulting PLCC for the proposed technique and also for two of the synthesis specific metrics considered previously, are presented in Table 5.29. These results show that the proposed metric does not perform so well in SIAT-based dataset as in IRCCyN database; as mentioned above, for this database the synthesized images have many compression artifacts, that reduces the perceptual impact of synthesis artifacts. In fact, the best metric to assess this database, is a conventional 2D metric, namely MSSIM.

Table 5.29 – Correlation results between the subjective and objective quality scores for SIAT-based database.

<table>
<thead>
<tr>
<th>Metric</th>
<th>DSQM</th>
<th>3DSwlM</th>
<th>VCIP2013</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLCC</td>
<td>0.7138</td>
<td>0.6751</td>
<td>0.6968</td>
<td>0.6707</td>
</tr>
<tr>
<td>SROCC</td>
<td>0.7321</td>
<td>0.6333</td>
<td>0.6595</td>
<td>0.6489</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.6699</td>
<td>0.7055</td>
<td>0.6859</td>
<td>0.6643</td>
</tr>
</tbody>
</table>

Since the results for the proposed metric are not consistent between the two assessed databases, some modifications were made to overcome this problem. As the proposed metric was developed using a database without compression artifacts, the parameter $\alpha$ naturally weights more the structural component of the metric, since this component evaluates better the synthesis artifacts. In section 5.4, it was shown that the PLCC between the subjective and the objective quality score were quite similar for a large range of $\alpha$ values; therefore, this parameter can be decreased, resulting in a higher weight for the non-structural component, without compromising the results obtained for the IRCCyN database. Table 5.30 and Table 5.31 present the new results for the IRCCyN and SIAT-based datasets, respectively, using $\alpha=0.5$. With this new setting, the proposed metric still outperforms all the considered metrics for the IRCCyN database, showing also a better performance for the SIAT dataset.

Table 5.30 – Metrics assessment results for the IRCCyN dataset, using $\alpha=0.5$ in the proposed solution.

<table>
<thead>
<tr>
<th>Metric</th>
<th>DSQM</th>
<th>3DSwlM</th>
<th>SSIM-based VSQA</th>
<th>VCIP2013</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLCC</td>
<td>0.6192</td>
<td>0.6864</td>
<td>0.6142</td>
<td>0.6447</td>
<td>0.7624</td>
</tr>
<tr>
<td>SROCC</td>
<td>0.4106</td>
<td>0.6156</td>
<td>-</td>
<td>0.5989</td>
<td>0.7106</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.5228</td>
<td>0.4842</td>
<td>-</td>
<td>0.5090</td>
<td>0.4102</td>
</tr>
</tbody>
</table>
Table 5.31 – Metrics assessment results for the SIAT-based dataset, using α=0.5 in the proposed solution.

<table>
<thead>
<tr>
<th></th>
<th>DSQM</th>
<th>3DSwIM</th>
<th>VCIP2013</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLCC</td>
<td>0.7138</td>
<td>0.6751</td>
<td>0.6968</td>
<td><strong>0.6944</strong></td>
</tr>
<tr>
<td>SROCC</td>
<td>0.7321</td>
<td>0.6333</td>
<td>0.6595</td>
<td><strong>0.6707</strong></td>
</tr>
<tr>
<td>RMSE</td>
<td>0.6699</td>
<td>0.7055</td>
<td>0.6859</td>
<td><strong>0.6445</strong></td>
</tr>
</tbody>
</table>

Figure 5.13 presents the scatter plot of the fitted metric values (MOSP or DMOSP) versus the subjective scores (MOS or DMOS) for new results for the SIAT-based dataset, using α=0.5.

![Scatter plots](image)

**Figure 5.13** - Scatter plot of fitted objective scores vs subjective scores referring to the SIAT-based dataset for the following metrics: a) DSQM; b) 3DSwIM; c) VCIP2013; d) Proposed.

As a final remark, these results indicate that instead of a priori setting a value for α, this value should change according to the amount of compression of the source images - if the source images have been more compressed, the value of α should result in a higher weight of the 2D Image Quality component, since it evaluates better the compression artifacts; the opposite should happen if the source images have not been compressed, where artifacts resulting from the synthesis procedure are more visible, and more weight should be given to the Structural Quality component.
Chapter 6. Summary and Future Directions

6.1 Summary

Recently, new applications based on 3D video have gained momentum, due to several technological improvements such as the display of 3D content with higher quality as well as low cost 3D acquisition equipment. Free view point video (FVV) is one of the 3D video applications which is currently emerging and still has high potential to be explored. However, for the full acceptance of FVV, a high user QoE should be guaranteed. This requires the development of objective video quality metrics for the DIBR-based synthesized images, allowing an accurate estimation of the perceived video quality.

The focus of the thesis was on the multiview plus depth 3D video representation, especially targeting free viewpoint (FVV) applications, which introduces new video artifacts resulting from the synthesis process, whose impact on the user perceived quality cannot be properly assessed with conventional 2D quality metrics. Accordingly, specific quality metrics for synthesized views have been recently proposed in the literature and some of them were reviewed in this thesis. Since the perceptual impact of the edges distortion resulting from the synthesis process was still not properly addressed in the literature, a new metric targeting mainly the edges distortion in synthesized images was proposed.

To evaluate the performance of the proposed metric, two databases were used: IRCCyN database and SIAT-based database. The first database was built only with synthesized images resulting from no compressed source images, and rendered using seven different DIBR algorithms, creating different synthesis artifacts; the second database was built with synthesized images resulting from compressed and uncompressed source images.

The metric performance evaluation was conducted using the Pearson correlation coefficient, the Spearman correlation coefficient and the RMSE between the metric objective scores and the subjective scores resulting from subjective tests. The results showed that the proposed metric performed well for the IRCCyN database, showing a good performance in quantifying the synthesis artifacts. On the other hand, the performance for the SIAT-based database was not so good; in fact, the compression artifacts (inherit from the source views) mask the synthesis artifacts, making it harder to identify them.

The performance of the proposed metric was also compared with other solutions existing in the literature, developed for conventional 2D images or for the specific case of synthesized images. The results showed that for the IRCCyN database, the proposed metric outperforms the other considered state-of-the-art metrics; for the SIAT-based database, and due to the compression artifacts, the 2D metrics (e.g. SSIM or MSSIM) perform better than the metrics developed specifically for synthesized images. In fact, this happens because the conventional 2D quality metrics are still the best metrics to evaluate of compression artifacts and the synthesis artifacts are masked.
6.2 Future Work

Despite the high performance of the proposed metric, in terms of predicting the quality of images with synthesis artifacts, there are some improvements and approaches that could improve the results, especially in the assessment of images with both synthesis and compression artifacts:

- Improve the estimation of the parameter $\alpha$ in the Fusion module. Instead of fixing a value for $\alpha$, this value could change according to the amount of compression used in the source images. If the source images have been more compressed, the value of $\alpha$ should result in a higher weight of the 2D Image Quality Metric, since it evaluates better compression artifacts. The opposite should happen if the source images have not been compressed, where artifacts resulting from the synthesis procedure are more visible, and more weight should be given to the Structural Quality metric.

- An improved database could be created to have a larger set of distortion artifacts, besides synthesis and compression related ones, on the images under evaluation. This enriched database could be used to apply a clustering algorithm to group the images with similar distortion types. Then, for each cluster, the right values of $\alpha$ and $\rho$ could be estimated, to give more weight to the metric which estimates better the predominant distortion type.

- The proposed metric can also be extended to video. In a first approach, the quality score for a video can be obtained by averaging the quality at each frame. In a second approach, the interdependency between frames can be explored, taking into account that the perceptual impact of the video is higher in areas of the scene where the action is happening.

All these improvements and different approaches can be relevant and interesting points for a future work, since their implementation could bring better results in a future metric.
Appendix A

This appendix includes the comparison of two approaches to detect the edges of an image: using a fixed threshold of $T=0.1$ or using an adaptive threshold which is selected automatically by the Canny edge function of Matlab. To compare the difference of the two approaches, one image was selected from each sequence of IRCCyN database and SIAT-based dataset.

**IRCCyN database**

Figure A.1 to Figure A.3 present an image for each video sequence of IRCCyN database and its respective edge detection using a fixed threshold of $T = 0.1$ and using the adaptive threshold.

![Image A.1](image1.png)

Figure A.1 – Image extracted from BookArrival sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.

![Image A.2](image2.png)

Figure A.2 – Image extracted from Lovebird sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.

![Image A.3](image3.png)

Figure A.3 – Image extracted from Newspaper sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.
SIAT-based Dataset

Figure A.4 to Figure A.13 present an image for each video sequence of SIAT-based dataset and its respective edge detection using a fixed threshold of $T = 0.1$ and using the adaptive threshold.

**Figure A.4** – Image extracted from Balloons sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.

**Figure A.5** – Image extracted from BookArrival sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.

**Figure A.6** – Image extracted from Dancer sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.
Figure A.7 – Image extracted from GT Fly sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.

Figure A.8 – Image extracted from Kendo sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.

Figure A.9 – Image extracted from Lovebird1 sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.

Figure A.10 – Image extracted from Newspaper sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.
Figure A.11 – Image extracted from Poznan Hall sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.

Figure A.12 – Image extracted from Poznan Street sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.

Figure A.13 – Image extracted from Shark sequence: a) original image; b) Edge image with $T = 0.1$; c) Edge image using the adaptive threshold.
Bibliography


[60] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and


