Image based representation for 3D content delivery

Chew, Boon Seng

2012

Chew, B. S. (2012). Image based representation for 3D content delivery. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/50767>

<https://doi.org/10.32657/10356/50767>

IMAGE BASED REPRESENTATION FOR 3D CONTENT DELIVERY

CHEW BOON SENG

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2012

Acknowledgments

I would like to express the deepest appreciation and sincere gratitude to my supervisor, Dr. Chau Lap Pui, for his guidance, kindness and many invaluable comments and suggestions throughout my Ph.D journey. Without his encouragement and support, the achievement of my thesis is impossible.

I would also like to take this opportunity to acknowledge my family members who have shown so much concern and care in the course of my studies. Their encouragement and unyielding support have contributed to the completion of this thesis.

I wish to thank Dr. Yap Kim Hui for his guidance and advices on the clustering algorithms for Virtual Human, Dr. He Ying, Dr. Steven C. Hoi and Wang Dayong on the research work together for the Spectral Geometry image.

Also, I would like to thank Nanyang Technological University for the award of a research scholarship which enabled me to undertake this research. I am grateful to my friends. Without them, I would feel much alone in this long journey.

Last but not the least, my special thank goes to the wonderful lady who has always been by my side throughout the ups and down of my studies. Susan, my beloved wife, thank you for your constant love, encouragement and understanding. Without your support, I would not have the strength to complete this thesis alone. Thank you.

Summary

The evolution of the Internet from a basic communication tool to a content driven system pushes the development of 3D virtual world interaction and exploration to a new height in recent years. This is a significant trend seen from the rapid growth in virtual world community and applications like 3D online gaming and content sharing. With the increasing demand for high quality 3D contents, interactive avatars and realistic virtual world environment, there is an urgent need to address the effective representation and smooth delivery of such information across networks given the erratic nature of the Internet and wireless channels.

This thesis focuses on the theoretical analysis and the introduction of efficient encoding techniques for static 3D models and dynamic Virtual Character Animation data. First, a new image format for the representation of 3D progressive model is proposed. The powerful spectral analysis is combined with the state of art Geometry Image (GI) to encode static 3D models into spectral geometry images (SGI) for robust 3D shape representation. Based on the 3D model's surface characteristics, SGI separates the geometrical image into low and high frequency layers to achieve effective Level of Details (LOD) modeling. For SGI, the connectivity data of the model is implicitly encoded in the image, thus removing the need for additional channel bits to be allocated for its protection during transmission. It is shown that by coupling SGI together with an efficient channel allocation scheme, an image based method for 3D representation suitable for adoption in conventional broadcasting standard is demonstrated.

Second, two encoding schemes that focus on the efficient representation and compression of the Virtual Character Animation (VCA) are shown. The proposed

Scalable virtual character animation (SVCA) demonstrates a new technique for transmitting level of details motion sequences for virtual character animation, focusing on the structural coherence characteristic of the virtual avatar. The virtual character encoder is scalable in nature, providing a form of flexibility for the bit-stream to be decodable at any bit rate to address the dynamic bandwidth constraint of a heterogeneous wireless network. In addition, the SVCA method exploits the structural behavior of VCA nodes and re-sequence the data packets based on their structural contribution to the reconstructed VCA to address the challenges of effective transmission.

Virtual Character Animation Map (VCAM) is introduced next to address the limitation in the SVCA. VCAM further improves the compression efficiency on the VCA data. It can be shown that by representing each segment of motion information of a skeletal avatar as an image map, temporal coherence presented within the joints sequences of the VCA can be exploited as spatial redundancies within an image. Conventional image processing tools and standards can then be applied to achieve efficient compression for the character animation sequences. The simulation results show that the proposed scheme achieves improved rate-distortion performance in comparison to the SVCA scheme for similar VCA sequences.

Third, a novel concept Virtual Character Animation Image (VCAI) is proposed. The VCAI jointly considers the structural and temporal coherence of the VCA to achieve compression and scalability of VCA data making it suitable for progressive transmission applications. Built upon a fuzzy clustering algorithm, the data similarity within the anatomy structure of a Virtual Character (VC) model is jointly considered with the temporal coherence within the motion data to achieve efficient data compression. Since the VCA is mapped as an image, the use of image processing tools is possible for efficient compression and delivery of such content across dynamic networks. A modified motion filter (MMF) is proposed to minimize the visual discontinuity in VCA's motion due to the quantization and transmission errors. The MMF helps to remove high frequency noise

components and smoothen the motion signal resulting in perceptually improved VCA with less distortion.

Lastly, a new technique based on the maximizing of Mutual Information for VCAI data is introduced. A systematic approach is used to model the inherent dependency between the structural characteristic and motion behavior of the skeletal model. The correlation based method is then used to achieve effective compression of the VCAI representation. The experimental performance of the new scheme in comparison to the VCAI for lossless encoding is shown and discussed in the thesis.

Contents

Acknowledgments	i
Abstract	ii
List of Figures	ix
List of Tables	xv
Table of Abbreviations	xviii
Table of Definitions	xxi
Table of Notations	xxiii
 1 Introduction	 1
1.1 Motivation	1
1.1.1 Limitations in polygonal mesh models	2
1.1.2 Image approach for encoding 3D Geometry	3
1.1.3 Emergence of Motion Capture	3
1.1.4 Limitations in encoding Virtual Character Animation	4
1.2 Objective	4
1.3 Major contributions of the Thesis	5
1.4 Organization of the Thesis	7
 2 Background	 10
2.1 Introduction	10
2.2 Polygonal 3D representation	11
2.2.1 Single Rate Compression	12
2.2.2 Progressive Compression	15

2.3	Image Based 3D representation	18
2.4	Virtual Character Representation	20
2.5	Progressive transmission	23
2.6	Conclusion	26
3	Spectral Geometry Images (SGI)	27
3.1	Introduction	27
3.1.1	Geometry Image and Spectral Analysis	29
3.1.2	3D content delivery	30
3.1.3	Contributions	30
3.1.4	Focus and Organization of Chapter	31
3.2	Review of JPEG2000	32
3.3	Geometry Encoding	33
3.3.1	Manifold harmonics transformation	33
3.3.2	Conformal parameterizations of 3D models to rectangular domain	36
3.3.3	Construction of spectral geometry image	39
3.4	Spectral geometry image compression	41
3.5	Channel coding	42
3.5.1	Channel Model	42
3.5.2	Problem formulation	43
3.6	JSCC for SGI	43
3.6.1	Distortion metric	47
3.7	Simulation Results	50
3.8	Discussion and comparison against state-of-art techniques for 3D representation	54
3.9	Conclusions	58

4	LOD and compression of Virtual Character Animation	61
4.1	Introduction	61
4.1.1	Virtual human representation	62
4.1.2	MPEG community for VCA development	62
4.1.3	Contributions	63
4.1.4	Focus and Organization of Chapter	64
4.2	VCA matrix representation	64
4.3	Euler Angles Representation	66
4.4	Quaternion	69
4.5	Bit-plane Coding	72
4.6	Scalable Virtual Character Animation	74
4.7	Virtual Character Animation Mapping	79
4.7.1	VCA decoded quality for varying motion frames length	81
4.7.2	Image map compression	81
4.7.3	Error metric	82
4.8	Experimental Simulation	83
4.9	Conclusion and Discussion	87
5	Clustering Approaches for Character Animation Representation	88
5.1	Introduction	88
5.1.1	VCA compression techniques	88
5.1.2	Contribution	89
5.1.3	Focus and Organization of Chapter	90
5.2	Proposed Framework for VCA representation	90
5.2.1	Temporal coherence within motion	91
5.2.2	Clustering Approach	93
5.2.3	VCAI compression and motion smoothening	96
5.2.4	VCAI mapping and Error metric	97
5.3	Experimental Result	99

5.3.1	VCAI using K-mean clustering	100
5.3.2	VCAI using Fuzzy-C mean clustering	102
5.4	Conclusions and discussion	107
6	Maximum Mutual Information for Virtual Character's Motion	
	Compression	110
6.1	Introduction	110
6.1.1	Review on VCAI and Correlations	110
6.1.2	Contributions	111
6.1.3	Focus and Organization of Chapter	112
6.2	Entropy and Mutual Information	113
6.3	Parameters for skeletal based animation	113
6.4	Determination of Maximum Mutual Information	114
6.5	Image Compression for VCA-MI	117
6.6	Experimental Results	119
6.7	Conclusions	123
7	Conclusions and Future Work	124
7.1	Conclusions	124
7.2	Future Work	126
7.2.1	Extension of SGI research	127
7.2.2	Progressive transmission of 3D animation using SGI	127
7.2.3	Error resilient transmission of 3D animated meshes	128
7.2.4	Segmentation of VCA for efficient motion analysis	128
7.2.5	Efficient representation and compression of VCA-MMI data	129
	Author's Publications	130
	References	132

List of Figures

1.1	Geometry image of the David head. Geometry image encodes the geometry x , y , and z into a regular grid of the colors r , g and b , which in general is continuous over the entire image domain. . .	2
2.1	Vertex, edge and face information for Figure-8 3D model.	11
3.1	2-level discrete wavelet transform and decomposition for image Lena into subbands.	32
3.2	Mesh diagram showing Laplace Beltrami operators.	34
3.3	Spectral analysis on 3D surface. The eigenfunctions of Laplace-Beltrami operator are orthogonal and serve the manifold harmonics basis. Row 1: the color indicates the function value. Row 2: the texture mapping shows the isocurves of the basis functions. .	35
3.4	Reconstructing the 3D mesh from the frequency domain. The number above each model is the number of eigenfunctions used in surface reconstruction.	35

3.5	Conformal parameterization of the genus-0 Bimba model. (a) The topology is first modified by two cuts, one at the top and the other at the bottom of the model. The cut surface M' is a topological cylinder. (b) The uniform flat metric by discrete Ricci flow is computed and M' is embed to a topological annulus. (c) Next, the topological annulus is mapped to a canonical annulus by a Möbius transformation. (d) The canonical annulus is cut by a line passing through the origin and conformally mapped to a rectangle. (e)-(f) The checkerboard texture mapping illustrates the conformality of the parameterization.	37
3.6	Spectral geometry image. (a)-(b) show the geometry image of the Bimba model. (c)-(f) show the 3-layer spectral geometry image. The normal maps in (b) and (d) highlight the difference between SGI_1 and GI . To better view the high frequency layers (e) SGI_2 and (f) SGI_3 , the pixel values are normalized to $[0, 255]$	38
3.7	JPEG2K vs JPEG-XR. One can clearly see that JPEG2K is more effective than JPEG-XR at low <i>bpp</i> , while JPEG-XR is more preferred for lossy compression of high quality results. Due to the smooth nature, geometry images have better measured PSNR compared to natural images from the experiment.	41
3.8	Packetization scheme. Figure demonstrates the packetization scheme and UEP allocation adopted for the SGI encoded model from the initial EEP configuration.	46
3.9	Conformal parameterization of the test models.	48
3.10	Mean curvature error d_H measures the visual quality. Row 1: the GI of resolution 256×256 ; Row 2: the 3-layer SGI of resolution 64×64 , 128×128 and 256×256 . The numbers below each figure are the <i>bits per pixel (bpp)</i> and the mean curvature error d_H . SGI has smaller d_H at low <i>bpp</i> because the high frequency layer is discarded and the low-frequency layer has less distortion than GI.	49

3.11	Shape compression using geometry image (GI) and spectral geometry image (SGI). GI is of resolution 256×256 for both models. The two SGIs with 3 layers are constructed respectively. Both Bunny and Foot models are of resolution 64×64 , 128×128 , and 256×256 . As shown in (a) and (b), SGI's performance is better than GI for varying compression ratio.	50
3.12	Comparison of error protection schemes. (a)-(b) show the mean curvature error for both (a)EEP and (b)UEP scheme with fixed source rate of 0.8bpp, 0.1-0.4bpp channel rate for the Bunny model. (c)-(d) show the mean curvature error for both (c)optimized EEP and (d)JSCC scheme with overall bit rate of 0.8-1.2bpp for the Bunny model. Horizontal axis denotes the packet lost rate P_{LR} in (%). The mean curvature error d_H is represented by $10\log_{10}(1/d_H)$.	51
3.13	Comparison of error protection schemes. (a)-(b) show the mean curvature error for both (a)EEP and (b)UEP scheme with fixed source rate of 1.2bpp, 0.1-0.4bpp channel rate for the Foot model. (c)-(d) show the mean curvature error for both (c)optimized EEP and (d)JSCC scheme with overall bit rate of 1.2-1.6bpp for the Foot model. Horizontal axis denotes the packet lost rate P_{LR} in (%). The mean curvature error d_H is represented by $10\log_{10}(1/d_H)$.	52
3.14	Comparison of d_H between JSCC allocation scheme for both GI and SGI method. (a)-(b) show the simulation results for (a)Bunny and (b)Foot model across varying packet loss rate. GI and SGI are both of resolution 256×256 . As shown, SGI outperforms GI for various packet loss condition. The number below the figure are the packet lost rate (%) and the vertical axis is the d_H denoted in $10\log_{10}(1/d_r)$	55

3.15	Discrete wavelet transform is highly dependent on the parameterization and re-sampling. (a) and (b) show DWT applied to 2D regular grid with different parameterizations. The middle and right figures show the reconstructed meshes with LL subband and LL+HL+LH subbands respectively. Clearly, the parameterization with poor quality results in the jaggedness artifacts due to high anisotropy. (c) Manifold harmonics transform is performed on the 3D meshes directly, thus, independent of the parameterization. From left to right, the reconstructed meshes with the number of eigenfunctions, 160, 500 and 1000 respectively. (d) shows the re-sampling of MHT results by the parameterization of poor quality. It has much less artifacts than that of (b).	60
4.1	Skeletal representation of the VCA model denoting the hierarchical characteristic between individual joints.	65
4.2	VCA of run and leap sequence 49_05 and running sequence 02_03. The motion sequence (a) and (b) consist of a total of 164 and 173 frames respectively. The original AMC files for (a) is approximately 128KB and (b)135KB in size.	65
4.3	VC table to summarize the representation for each joints id and its corresponding DOF limitation. Initial 3 DOF denotes the position of the root of the skeleton. Remaining i-3 DOF of the VC consist of the Euler angles needed to specify each joint position.	67
4.4	Euler rotation using Z-Y-Z.	69
4.5	Bit-plane Coding technique example.	73
4.6	Reconstruction of DOF-50 with progressive increment of bit-plane 1-4.	75
4.7	Reconstruction of DOF-50 with progressive increment of bit-plane 4-16 layers.	76

4.8	Comparison of DEF vs data size between BP-VCA and SVCA for run and leap VCAs sequence.	78
4.9	For both images, the motion information is encoded as a JPEG2000 [133] image before compression where $m=62$, (a) demonstrates the image mapping for run and leap sequence and (b) shows the image mapping for cartwheeling sequence. Both image map exhibit the characteristic of strong temporal coherence along the row entity.	80
4.10	The sequence of image maps denote motion of the skeletal avatar for the walking animation. The motion data consist of 652 frames of motions and $m=62$ in this example. Individually, each mapping is normalized and compressed using JPEG 2000 encoder [14].	80
5.1	VCAI encoding process with motion smoothening. (a) shows the original DOF trajectory values against frame number for 02_07 sequence. (b) shows the image mapping of DOF values to grayscale image. (c) denotes the VCAI which consider both anatomical structure of VC and motion coherence. (d) shows the VCAI with MMF for motion smoothening.	92
5.2	Comparison results for the lossy compression of VCAI technique for sequence (a-b)02_01 and (c-d)02_03.	103
5.3	Comparison results for the lossy compression of VCAI technique for sequence (a-b)49_05 and (c-d)49_06.	104
5.4	Comparison results for the lossy compression of VCAI (MMF) against VCAI for sequence (a-b)02_01 and (c-d)02_03 are reported. The R-D plot from low to high Cr is presented in the figures. (a) and (c) denote the results from the original VCAI (FCM) method, (b) and (d) represent the improved experimental results with the VCAI (MMF).	106

5.5	Reconstructed frames on varying motion sequences with the VCAI scheme. The individual sequence are clustered based on their DOFs similarities using FCM and compressed using the J2K standard. Both the motion temporal coherence and skeletal structure of the VC are exploited to improve compression efficiency and visual performance.	109
6.1	Framework for image segmentation of VCA data. The original VCA data is partitioned for both lossy and lossless compression. Segmentation for lossy components is performed to improve correlation of VCA sequence.	118

List of Tables

4.1	Results for the compression performance of the VCA mapping technique presented. Here, FN - original motion capture filename, Desc - the description of the motion sequence, AMC - the AMC file size in (KB), Fr - number of frames in sequence, Cs - Compressed file size, CR - compression ratio of uncompressed to compressed motion files.	84
4.2	Results for the lossy compression of VCA mapping technique for sequence 49_05 is reported. Here, R - denotes the compression ratio used in JPEG 2000 standard, AMC - original motion data file size in (KB), Cs - the compressed file size of the motion file (KB), DEF - Displacement Error per frame, CR - compression ratio of uncompressed to compressed motion files.	84
4.3	Results for the lossy compression of VCA mapping technique for sequence 02_07 is reported. Here, R - denotes the compression ratio used in JPEG 2000 standard, AMC - original motion data file size in (KB), Cs - the compressed file size of the motion file (KB), DEF - Displacement Error per frame, CR - compression ratio of uncompressed to compressed motion files.	85

4.4	The table shows the rate distortion comparison between the proposed scheme compared to its SVCA counterpart for the run and leap sequence. In the experiment, the results from the individual compression scheme is presented and compared. Here, Cs denotes the compressed file size of the motion file (KB), DEF - Displacement Error per frame and CR - compression ratio of uncompressed to compressed motion files.	86
5.1	Results for the compression performance of the VCA mapping technique using k-mean clustering. Here, FN - denotes the motion index, Desc - denotes the motion description, AMC - original motion data file size in (KB), Fr - number of frames, Ns - denotes number of segment, Cs - the compressed file size of the motion file (KB) using J2K, DEF - Displacement Error per frame, CR - compression ratio of uncompressed to compressed motion files. .	100
5.2	Results for the compression performance of the VCA mapping technique using FCM. Here, FN - denotes the motion index, Desc - denotes the motion description, AMC - original motion data file size in (KB), Fr - number of frames, Ns - denotes number of segment, Cs - the compressed file size of the motion file (KB) using J2K, DEF - Displacement Error per frame, D_{norm} - Normalized Distance, CR - compression ratio of uncompressed to compressed motion files.	101
5.3	Results for the performance of un-clustered VCAI (Cs_1 , DEF_1 , $Dnorm_1$ and Cr_1 vs FCM technique(Cs_2 , DEF_2 , $Dnorm_2$ and Cr_2)).	102

6.1	Root mean square(RMS) errors of different test motion capture data using the VCA-MI approach. Here, <i>Orig_Size</i> (KB) - original motion capture AMC file size in (KBytes), <i>Segs</i> - number of segments, <i>Comp_Size</i> - Compressed file size of VCA data, d_r - Root mean square Error, <i>Cr</i> - Compression ratio	120
6.2	Results for the VCA-MI performances across varying compression rates. Here, <i>RMS</i> denotes the Root mean square error. <i>Q</i> is the varying compression rate value of J2K encoder. <i>Cs</i> and <i>Cr</i> are the Compressed file size(KBytes) and Compression ratio respectively.	121
6.3	Comparison of VCA-MI against VCAI for lossless compression. Here, <i>DEF</i> denotes the displacement error per frame and <i>RMS</i> denotes the Root mean square error.	121

Table of Abbreviations

ASF/AMC Acclaim Skeleton/Motion Capture file

BP-VCA Bitplane Virtual Character Animation

CPM Compressed Progressive Meshes

CPCA Clustered Principal Component Analysis

DCT Discrete Cosine Transform

DEF Displacement of Error per frame

DMB Digital Multimedia Broadcasting

DOF Degree of Freedom

DVB-H Digital Video Broadcasting- Handheld

DWT Discrete Wavelet Transform

EBCOT Embedded Block Coding with Optimized Truncation

EEP Equal Error Protection

FCM Fuzzy C Means

FEC Forward Error Correction

GI Geometry Image

GPS Global Point Signature

GPU Graphics Processing Unit

HCI Human Computer Interaction

J2K JPEG 2000

JPEG-XR JPEG Extended Range

JSCC Joint Source and Channel Coding

LOD Level of Details

MDC Multiple Description Coding

MHT Manifold Harmonics Transform

MoCap Motion Capture

MPEG-4 AFX MPEG-4 Animation Framework eXtension

MPEG-4 BAP MPEG-4 Body Animation Parameters

MPEG-4 BIFs MPEG-4 Binary Format for Scene

MPEG-4 SNHC MPEG-4 Synthetic Natural Hybrid Coding

PCA Principal Component Analysis

PM Progressive Mesh

PFS Progressive Forest Splitting

PSC Progressive Simplicial Complexes

QEM Quadric Error Metrics

RS code Reed Solomon Code

ROI Region of Interest

RMS Root Mean Square

SGI Spectral Geometry Image

SVC Scalable Video Coding

SVCA Scalable Virtual Character Animation

TG Coder Touman and Gotsman Coder

UEP Unequal Error Protection

VC Virtual Character

VCA Virtual Character Animation

VCA-MMF Virtual Character Animation (Modified Motion Filter)

VCA-MI Virtual Character Animation (Mutual Information)

VCAI Virtual Character Animation Image

VCAM Virtual Character Animation Map

VRML Virtual Reality Modeling Language

Table of Definitions

Channel Coding:	Technique of applying error control methods to protect or recover data due to loss in noisy communication channels.
Clustering:	Assignment of data into groups based on similarity or correlation studies.
Connectivity Information:	Information use to describe the relationship between polygons or triangle necessary during the reconstruction of the 3D model.
Correlation:	Statistical similarities between two dependent/independent sets of data or variables.
Degree of Freedom:	Displacement or rotation parameters to specify the motions of virtual human motion across time.
Diffeomorphism:	Smooth bijective mapping from one function to another and reversible in nature.
Dynamic 3D model:	Alternative definition for 3D animation. 3D models with the capacity to change across time.
Equal Error Protection:	Equal allocation of FEC bits across data bitstream.
Forward Error Correction:	Technique use in digital communication for error correction through introduction of redundancies within bitstream.
Geometrical Information:	Positional information of vertices within the the 3D model.

Image Parameterizations:	Mathematical representation and transformation of 3D model to image representation.
Level of Details:	Representation of 3D content encoded to different resolutions and details.
Lossless Compression:	Reduction of data size without physical loss of information.
Lossy Compression:	Reduction of data size through physical discard of information.
Motion Capture:	Recording of complex human's motion using specialized equipments, common in cinematic industries.
Mutual Information:	Statistical measurement of dependency/correlations between two set of variables.
Source Coding:	Compression of 3D content.
Spectral Geometry:	Field of studies for relationship between geometrical surfaces of 3D models against differential operators.
Static 3D model:	3D models without temporal changes across time.
Virtual Character:	Skeletal Based virtual human representation obtained from motion capture techniques.
Unequal Error Protection:	Unequal allocation of FEC bits across data bitstream based on importance of layers.

Table of Notations

Summary of Key notations used in different Chapters.

Chapter 3

a_β	Sub-band layer
q_β	Quantized Sub-band layer
Δ_β	Quantization Step Size
Δ_{ij}	Laplace Beltrami operators
A_i, A_j	Areas of triangles that share edge v_i, v_j
(H^k, λ_k)	Eigenfunction and Eigenvalues pairs
D	Rectangular domain where $D \in \mathbb{R}^2$
M, M'	Original 3D model and cut surface
$\phi(M')$	Embedded Surface
$\phi(C_1), \phi(C_2)$	Inner and outer circle of topological annulus
M_i	Reconstructed mesh with m_i eigenfunctions
Q_{budget}, R_{s+c}	Total bit budget, coding rates for Source and Channel coding
Q_r	Compression ratio for i layer using JPEG2000
d_H	Average of norm of geometric and Laplacian differences

Chapter 4

μ_i^j	VCA data, j indicates frame number, i denote DOF
m, n	Total frames and total joints
M_R	Rotation matrix
Q	Quaternion Set for Eqn. 4.7-4.12
C_i^j	Transform coefficient of DOF i
L_i	Layered i generated from BP-VCA, L_0 is defined as the base layer
DEF	Displacement error per frame
$\gamma(J_{\beta j}), \emptyset(J_{\beta j})$	decoded joint positions of VCA, original joint position of VCA
C_s	Compressed data size of J2K image
R	Compression ratio in JPEG2000
CR	Compression ratio of compressed VCA against original AMC

Chapter 5

μ_i^j	VCA data, j indicates frame number, i denote DOF
α_x	Centroid for k-mean clusters
S_x	Clustered Set where $S_x = (S_1, S_2, \dots, S_k)$
n_x	number of DOF for cluster
ψ_β	Graded coefficient, measurement of membership to the cluster
α_β	Centroid for FCM clusters
C_s	Total compressed data for VCA sequence using J2k image, $C_s = \sum_i^k C_i$
$C_1, C_2, \dots, C_k$	Individual J2K image needed to decode the VCA sequence
$\gamma(J_{\beta j}), \emptyset(J_{\beta j})$	decoded joint positions of VCA, original joint position of VCA
D_{norm}	Normalized distance
C_r	Compression ratio of compressed VCAI against original AMC data

Chapter 6

μ_i^j	VCA data, j indicates frame number, i denote DOF
X, Y	Lossless DOF matrix, Lossy DOF matrix
ξ	Complete DOFs to decode full VCA sequence
$\psi(Q)$	Entropy for the Qth DOF
$p(q_\tau)$	probability function
$MI(Q S)$	Mutual information between set Q and S
NI	DOF index
Q	Compression ratio used in J2K
d_r	Root mean square error measurement
C_r	Compression ratio against of compressed VCA against original AMC data
C_s	Compressed VCA data in image form

Chapter 1

Introduction

1.1 Motivation

The migration of two dimensional, natural and synthetic image to a 3D representation can be considered as a natural evolution in the world of graphics, to satisfy human desire for realism and interaction with a virtual creation. With the rapid growth of technology in scanning devices, computer processor and graphics processing unit(GPU), the acquisition and processing of the conventional 3D models become simplified. In the current context, the use of 3D content can be found across diversified fields such as education, medical, robotics, navigation and engineering simulation. Recently, there is an increasing demand for such content in the entertainment sector due to the emerging market of 3D animatics movies and gaming. The use of 3D models offers many advantages over conventional 2D image. This includes providing the user with an additional sense of depth to understand the subject, and allowing greater flexibility in the manipulation and viewing of data content. However, the use and sharing of such 3D contents could often lead to challenges in the form of efficient representation, data storage, processing and transmission of the desired information.

Figure 1.1: Geometry image of the David head. Geometry image encodes the geometry x , y , and z into a regular grid of the colors r , g and b , which in general is continuous over the entire image domain.

1.1.1 Limitations in polygonal mesh models

Conventionally, most of this information is represented in the standard 3D dimensional format. Although much research was done to find more effective forms to describe 3D models, polygonal mesh is still the most commonly adopted data type for the representation and delivery of a 3D model/object in many real-world applications. The delivery of 3D contents using polygonal meshes have several limitations which are yet to be thoroughly researched for its wide implementation in practical systems. These include, efficient encoding and delivery techniques for 3D model connectivity data to ensure correct decoding of models, effective LOD schemes to enable progressive transmission of 3D content, improved compression performances and better model representation techniques to further reduce the data storage and processing requirements for highly complex 3D models and ensuring the processing and transmission of such contents are capable of meeting the stringent demand in real-time streaming applications.

1.1.2 Image approach for encoding 3D Geometry

Geometry image, introduced by Gu et al [60], provides an alternative way for shape representation and transmission. In sharp contrast to the irregular polygonal meshes, geometry image is a completely regular representation that all connectivity information is implicitly encoded in the image space. Furthermore, other geometry and appearance properties, such as normals, textures, materials, can also be stored using the same parameterization. Geometry image naturally bridges two research fields, image processing and geometry processing, and provides a way to borrow the well-studied image processing techniques to geometry processing. However, geometry image is fundamentally different from natural images. Geometry image in general is continuous and smooth over the entire image domain as shown in Fig. 1.1. To integrate geometry image with the existing image processing framework, there is a need for analyzing the geometric properties in geometry image space. This thesis explores such challenges and seeks to serve this purpose through the introduction of the new concept called *Spectral Geometry Image (SGI)*.

1.1.3 Emergence of Motion Capture

Motion capturing is fast becoming a common norm for synthesizing complex human motion into skeletal virtual avatar. Recently, there is a trend for the increasing usage of Virtual character animation in the domain of human-computer interaction, movies, digital 3D animation, video games and low-powered mobile devices which require virtual avatar representation. There are several advantages

in using the motion capturing for realistic human motion simulations in comparison to the conventional graphics modeling and computer animation method. The motion capture offers rapid generation of complex human motions and enables accurate human anatomical behaviors to be replicated on virtual entities.

1.1.4 Limitations in encoding Virtual Character Animation

The current MPEG-4 body animation parameter (BAP) compression technique for virtual human sequences has yet to fully address the issue of delivering such content across the network in a progressive manner. Coupling the problem with an absence to jointly exploit the structural and temporal coherency of the virtual human data during lossy compression, unacceptable quality degradation of the character animation can occurred during the transmission of such information across the network. The commonly use, pre-obtained motion capture sequences exhibit two distinct characteristics in the form of structural coherency and temporal correlations. In the current context, most compression methods focused on either characteristic to achieve lossless/lossy data reduction while considering the perceived quality loss in the decoded VCA. This thesis seeks to address such limitations and proposes an effective scheme which jointly considers the unique structural and temporal characteristics of the VCA to achieve compression. The proposed technique is also scalable in nature and capable of addressing the problem of progressive transmission for the VCA data.

1.2 Objective

This thesis aims to further investigate more effective representation and compression techniques for emerging 3D content. The two key areas of focuses are

3D static model and Virtual Character Animation data. The main objectives of the thesis are summarized as follows.

- To investigate an image based approach using Spectral Geometry image to encode real-world 3D models.
- To present a complete framework for both source and channel encoding to deliver the static 3D model across lossy communication channels.
- To present the quantitative studies in comparison to state-of-art image based encoding techniques for 3D geometry.
- To investigate efficient representation and compression approach for the emerging virtual character animation.
- To exploit the unique skeletal motion characteristics to achieve a framework for constructing scalable VCA representation with high compression ratio performances and low quality degradation across varying bit rates.

1.3 Major contributions of the Thesis

The major contributions of the thesis are as follow:

- A new concept of Spectral Geometry Image (SGI) is presented. The framework for constructing spectral geometry images of real-world 3D models is developed. It is shown that spectral geometry image is more powerful and flexible than the conventional geometry image for 3D shape representation and compression. The joint source and channel coding scheme (JSCC) is coupled in the SGI framework to facilitate the transmission of static 3D

content in a lossy network environment. The framework ensures the graceful degradation of decoded 3D static model in comparison to conventional equal error protection (EEP) and unequal error protection (UEP) techniques across simulated packet loss channels. Simulation results presented the objective and subjective performances of the SGI.

- The SVCA is introduced to provide scalability and compression for the VCA data. The method focuses on the aspect of structural characteristics of the VCA data to assign source bits dependent on the importance of the degree of freedom (DOF) parameters. The SVCA is scalable to different level of details (LOD) based on the available bandwidth. The comparison against the non-assigned bit-plane virtual character animation (BP-VCA) is demonstrated in the simulation.
- VCAM further improves the compression performance of the human skeletal data in comparison to SVCA. The image based encoding scheme solely focuses on the compression of the VCA using temporal characteristic of the skeletal's motion. The proposed scheme made use of coherency within the human motion to achieve effective data reduction in the model's representation using conventional image processing technique. Simulation results show that the VCAM has better compression performance when compared against the SVCA method.
- The novel concept of VCAI is developed to address the limitations in both SVCA and VCAM techniques. The proposed VCAI method make use of a fuzzy clustering algorithm, to jointly consider both the data similarity

within the anatomy structure of a Virtual Character model and temporal coherence of VC motions to achieve efficient data compression. The simulation results demonstrated that the VCAI techniques show improved compression performance against state-of-art VCA encoding methods. Both objective and subjective evaluations of the VCAI techniques are provided in the thesis and online video demonstration.

- A compression approach for Virtual Character Animation (VCA) based on the criterion of maximization of mutual information (MMI) in complex human motion sequences is demonstrated. The proposed scheme uses the concept of MI to provide a qualitative measure towards determining the inherent dependency between structural and temporal information presented in the skeletal human motions. Experimental results show that the proposed VCA-MI is effective in generating subjectively pleasant VCA sequences across varying compression ratio for skeletal motion sequences of different complexity.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows.

Chapter 2 outlines the background for the related researches in 3D representation and compression. Single rate compression techniques for both the connectivity encoding and geometry compression were first reviewed to provide the necessary background for understanding the development during the earlier studies on 3D contents. The later sections of the chapter cover the more recent topics

on the progressive encoding of 3D static model, image based encoding schemes and delivery of model representation over lossy communication network. The review for the emerging studies for better representation and encoding techniques of the motion capture(Mocap) data is included in the later portion of the chapter.

In Chapter 3, the challenge in finding an efficient framework for the delivery of 3D static content over the lossy communication network is investigated. The Spectral Geometry Image is introduced as a solution to resolve the problem faced for the lossless transmission of connectivity information, and providing the flexibility for the 3D static model to be represented in different level of details, dependent on the available bandwidth of the communication channels. The channel encoding scheme for efficient delivery of the SGI content across packet loss network was covered in the later sections of the Chapter. Lastly, the simulation results and discussion for the SGI in comparison to the current state-of-art techniques are presented to demonstrate the advantage of the SGI framework.

Overview for the increasing usage and demand for the emerging Virtual Character Animation data is presented in Chapter 4. The Chapter introduces two independent VCA encoding schemes that focus on the different characteristics of the skeletal model type to achieve efficient compression and scalable encoding. The matrix representation and encoding process for the Scalable Virtual Character Animation and Virtual Character Animation Map were provided. Simulation and experimental comparison showed that the VCAM outperforms SVCA for compression performance.

In Chapter 5, Virtual Character Animation Image is introduced to address the limitations in both the SVCA and VCAM techniques. The framework to jointly

exploit the structural and temporal correlation for achieving efficient compression is discussed. The experimental comparison of the two clustering approaches for both the k-mean clustering and Fuzzy-c mean is presented to demonstrate the differences in the two performances. Simulation results of the VCAI and discussions against the state-of-art techniques are provided in the chapter.

In Chapter 6, the VCAI is combined with the Mutual information criterion to address the challenge of finding a systematic measure between the correlation within the structural VC data and temporal coherence of the human skeletal motion. An overview for the entropy encoding and mutual information concepts was covered in the earlier portion of the chapter. The algorithm for the maximizing of the MI is introduced. The simulation result shows that the VCA-MI technique is competitive against VCAI for lossless encoding performances.

This thesis is concluded in Chapter 7. Some recommendations for future research are also discussed in this chapter.

Chapter 2

Background

2.1 Introduction

This chapter reviews the related research done on 3D content representation, compression and transmission, to provide the necessary background and understanding for the following chapters of the thesis. The rest of the chapter is organized as follows. Section 2.2.1 discusses the earlier works which addressed the compression of 3D model using single rate compression techniques. The shift in focuses of research from the domain of lossless connectivity encoding towards geometry compression is prominent as the bit-rate for connectivity encoding pushes towards theoretical optimality. Section 2.2.2 reviews the related works for progressive compression. The rapid growth of the Internet plays a key role towards the development of 3D model representation to be more oriented towards transmission applications. Progressive compression techniques address the limitation for the need of full encoding and decoding of 3D-model in single-rate compression and allows the flexibility for partial compression of the 3D content to different resolution for transmission. Image based compression techniques are reviewed in Section 2.3. The section discusses the relevant spectral analysis technique and

Figure 2.1: Vertex, edge and face information for Figure-8 3D model.

conformality methods used for 3D surface analysis and 3D to 2D encoding. With the social media gaining pace and the utilization of virtual avatars increasing across media, several researchers [26][36][58][104] have looked into the dedicated representation and compression techniques for virtual skeletal data. This is covered in Section 2.4. Finally, the works which addressed the effective delivery of 3D representation across dynamic network constraints are discussed in Section 2.5.

2.2 Polygonal 3D representation

3D models are conventionally represented using the polygonal/triangular meshes which are commonly used in the graphics industry. The 3D polygonal mesh composed of a combinational key attributes of vertex, face and edge. The vertex information describes the individual point positions of the 3D model in R^3 space denoted by a set of $Vertex(V) = (V_x, V_y, V_z)$. The edge information defines the line segment which connects two vertex points at an instant. The face information

listed the relationship between the different line segments and vertex points to define the polygons in the 3D mesh. Figure 2.1 shows the relationship between vertex, edge and face information of a 3D model. The detailed and complex models used in graphics and simulation applications often requires large data size due to the huge amount of triangles and connectivity information needed to effectively encode the fine details of complex shapes.

2.2.1 Single Rate Compression

Connectivity Encoding Several early researches focused on single rate [25] and connectivity compression techniques to reduce the data size. This was done either through the exploitation of redundancies in the description of data or using remeshing techniques. Edgebreaker [113] which falls into the first category of exploiting redundancies, generates a descriptor for the topological relationship between the triangle and boundaries of the remaining mesh during traversal. It ensures a guaranteed worst case bound of 4bpv (bits per vertex) for simple meshes. The algorithm is limited for offline usage and not applicable for streaming application due to its two-pass process for decompression and $O(v^2)$ decompression time needed for decoding the model. Touma and Gotsman [135] alternatively proposed a valence driven approach which uses the characteristic of ordered vertices in an orientable and manifold mesh to further reduce the bpv for generic meshes to 2-3.5 and less than 0.2 bpv for highly regular meshes. Alliez and Desbrun [24] improved the performance in [135] by introducing a new valence-driven conquest for arbitrary meshes. They demonstrated the performance of their algorithm is upper-bounded by 3.24 bpv for large arbitrary meshes. The

entropy studies to prove that their algorithm is asymptotically optimal is shown in [24].

Geometry Compression-Quantization Aside from connectivity compression which are generally lossless in nature and requires less bits, Geometry compression techniques offer more room for data exploitation and capacity for better compression performances. The geometry information are generally encoded in floating representation where some precision loss can be tolerated by applications to achieve a higher compression ratio. The initial step to achieve this precision loss involves the *Quantization* process. Deering first introduced the term of Geometry Compression in [49] and proposed a generalized triangle mesh which allows 3D triangle mesh to compress and achieve a compression ratio of 6-10 to 1. [129] proposed a compressed binary format for virtual reality modeling language (VRML) based on the use of topologically assisted compression. Both [49][129] uniformly quantized the coordinates of the vertex positions in Cartesian space using 8-16 bits. As a benchmark, the precision of 16-bit integers are sufficient to represent 15μ details on the human body and 1mm details on large buildings as stated in [28]. Chow [46] presented a general meshifying algorithm to decompose a given triangulated mesh to generalized mesh and proposed the compression of different precision depending on the level of details on each regions of the generalized mesh. The work showed an improvement of 2 to 3 times compression performance over the technique proposed by [49]. Sorkine [128] demonstrated the problem of noticeable surface modification caused by high frequency error which is introduced by conventional quantization methods when applied directly

to Cartesian coordinates of the model. To overcome this, [128] manipulated the 3D model in frequency domain and quantized the geometry of the model based on concentration of quantization errors at the low frequency spectrum and preserves the fine normal information.

Predictive Encoding Similar to the ideology used in image/video compression [108], the vertex position can be further compressed using both predictive and entropy encoding techniques. A simple Delta coder is proposed by Deering [49] where quantized information such as positions, colors and normal information are first delta coded to exploit the correlations between neighboring vertices. The prediction errors or deltas are subsequently encoded with a modified Huffman coder to achieve data reduction. Chow [46] adopted similar delta coding method to encode the vertices and normal data and reported a bitrate of 13-17.5 bpv for quantization of 8.7-12 across the entire model. Taubin and Rossignac [132] used a vertex spanning tree to predict the position of next vertex based on a linear combination of previous vertices in a spanning tree. Their approach differs from [49] by exploiting the geometrical coherence of several ancestors in the spanning tree before entropy encoding the correction vectors. The delta coding method [49] is considered a subset of the generalized linear prediction method as earlier demonstrated in [132].

The application of parallelogram rule for geometry compression was first inspired in [135]. The TG coder uses the vertices of neighboring triangle to predict the position of a vertex based on the concept of forming a complete parallelogram. Alliez [76][79] generalized the parallelogram technique in TG coder and apply it

for polygonal meshes compression . Their code shows an improved performance over [135] with an average bitrate of 7.2-16.6 bpv for 8-12 bits precision. One limitation in their method is the dependency on characteristics of the polygon. In order to obtain a good prediction of the vertex position, the polygons have to be fairly planar and convex.

2.2.2 Progressive Compression

The 3D models encoded using the single rate compression techniques are often unsuitable for layered or progressive transmission. Since its encoded 3D model cannot be partially decoded at any instance during transmission, the full transmission of the entire 3D content is often required before rendering can be performed to retrieve the original model. This can lead to an unacceptable latency time which is especially crucial in real-time streaming applications. Progressive compression techniques provide the flexibility for a 3D model to be presented in different levels of detail to cater for variations in bandwidth and processing power capabilities.

Hoppe first introduced the use of progressive meshes (PM) [71] and inspires the area of research for progressive compression. The PM consists of a series of *vertex splits* to progressively improve the quality of the 3D model as more refinements are received at the decoder. The reverse process is *edge collapse* which removes vertex. The PM technique enables flexibility during transmission. However, there is a trade-off of larger data size and overhead information needed to explicitly encode the individual vertex location. The improved version of PM was proposed by Hoppe in [72] to enhance the compression performance of

PM through re-ordering of vertex split operation and introducing more efficient data structure. The technique for Progressive Simplicial Complexes (PSC) was introduced in [102] to overcome the two limitations of PM [71], (1) Limited applications to orientable 2-dimensional manifolds and (2) Lacking of flexibility to enable changes to the topology during coding. The PSC carried out generalized vertex splits to encode changes for both geometry and topological information of meshes. The successive refinements were then applied to a base model which consists of a single vertex to enable both the topological and geometry quality of the 3D meshes to be increased progressively. Gandoin and Devillers [55][56] applied the technique in PSC and used an earlier proposed Kd-tree geometric coder [51] to improve the joint compression performances of the positions and connectivity of the mesh. The proposed progressive geometry compression scheme is applicable for manifold meshes, triangle soups and 3D tetrahedral meshes.

The Progressive Forest Split(PFS) was proposed by Taubin et al [131] to enable a tradeoff of granularity in refinements against complexity of data stream. It was included in MPEG-4 3DMC [132] as part of the standard for progressive compression of 3D mesh. The method transmits the LOD hierarchies in a progressive and compressed form and is also applicable for manifold meshes. Similar to PM, the PFS is a progressive compression technique which carries out refinement operations first on a low resolution base model and subsequently improves the quality of the 3D model as more data is decoded. The PFS uses forest splits to cut the mesh along the edges of a forest which is similar to the grouping of several consecutive vertex split operations in PM to increase coding efficiency.

Pajarola and Rossignac presented the use the Compressed Progressive Meshes (CPM) [96] as a modified solution over PM to reduce the overhead and data size required for the shape representation of 3D meshes. The CPM algorithm groups and processes the vertex refinement in batches, each corresponding to a LOD layer. This results in the reduction of latency time for a 3D model to be partially rendered and previewed. CPM uses a reverse variant of edge-split butterfly subdivision scheme [53][148] to predict the unknown vectors from known vertices. The prediction errors are subsequently encoded using entropy coder [74]. CPM reported a result of 3.6 bits connectivity and 7.7 bits for vertex location using the bunny model. PM and PFS have shown to require 7 bits and 4.4 bits for connectivity and 15-25 bits and 18.9 bits for geometry encoding respectively.

Corhen-Or et al [48] further improved the encoding bit rate by introducing the use of a patch coloring algorithm. The vertex decimation is based on the 4-coloring and 2-coloring technique which depend on the distribution of patch degree in a given LOD. Similarly, [80] introduces the concept of progressive geometry compression (PGC) and compared their results against CPM. PGC uses semi-regular meshes for 3D models to significantly reduce the data budgets required for connectivity and parameter information. Coupled with wavelet and zerotree coder, their algorithm reported improved performance against the [96][132][135]. The valence driven and conquest approach was introduced by Alliez and Desbrun [23] for the purpose of lossless transmission of fine granularity triangle meshes. The decimation conquest algorithm via a combination of valence driven decimation and adaptive retriangulation ensures the valence

regularity during progressive encoding. Their algorithm reported a 2-5bpv for connectivity and 10-16.5bpv for geometry encoding.

The quantifying of visual quality of the reconstructed 3D model plays an important role in determining the effectiveness of the compression algorithm. Cignoni et al presented the METRO tool [47] which compares the differences in two 3D models based on point to surface measure. A similar scheme with improved speed was demonstrated in [27] based on the approximation of Hausdorff distance measure. Quadric Error Metric (QEM)[57] was introduced to achieve effective surface simplification and is capable of handling non-manifold 3D objects. The QEM associates a sets of triangle planes with each vertex to determine the squared error from each vertex. The quadric error is then applied during the simplification process to track the errors.

2.3 Image Based 3D representation

Geometry image [60] is an effective solution for compressing shapes that are parameterized to the regular domain, like the rectangle or sphere. Peyré and Mallat presented geometric bandlets to compress geometrically regular objects (like geometry images and normal maps) [100]. They showed that bandeletization algorithm removes the geometric redundancy of orthogonal wavelet coefficients and thus is more effective than the wavelet based compression. Lin et al considered the use of JPEG2K for compression and delivery of a geometry image [87]. They made use of the ROI (region of interest) characteristic in JPEG2K to achieve view dependent streaming. However, the impact of channel loss towards the decodable bit-stream of the 3D geometry model was not discussed in [87].

Spectral geometry processing relies on the eigenvalues and eigenvectors from mesh operators to carry out desired tasks. Motivated by the similarity of the eigenvectors of the graph Laplacian and the discrete Fourier transform [109][110], Taubin reduces the surface smoothing problem to low-pass filtering of the discrete surface signals [130]. Since then, there is a large amount of work in spectral geometry processing. A recent survey for such studies was summarized in the State of The Art Report (STAR) by Zhang et al [146].

Lévy pointed out the eigenfunctions of the Laplace-Beltrami differential operator capture the global properties of the surface, in some sense, “understand” the geometry [84]. Vallet and Lévy derived a symmetric discrete Laplacians using discrete exterior calculus which guarantees the eigenfunctions form an orthonormal basis, called manifold harmonics basis, with positive eigenvalues [138]. They also developed an efficient and numerically stable approach to compute the eigenfunctions of the Laplacian for meshes with up to a million vertices.

Manifold harmonics provide a natural way to analyze the signals defined on surfaces of arbitrary topology in a Fourier-transform like fashion, which thus has a wide range of applications in geometry processing. Rustamov presented the Global Point Signature (GPS), a deformation invariant shape signature using the eigenvalues and eigenfunctions of the Laplace-Beltrami operator [116]. Rong et al [112] proposed spectral mesh deformation that compactly encodes the deformation functions in the frequency domain. Liu et al [90] presented a robust, blind, and imperceptible spectral watermarking approach for polygonal meshes using manifold harmonics transform. They demonstrated that the spectral ap-

proach is very promising to be robust against noise-addition and simplification attacks.

Another related work is the spectral coding algorithm presented by Karni and Gotsman [77]. First, it partitions the 3D model into several submeshes, then computes the spectral of the adjacency matrix for each submesh. Finally, the spectral coefficients are quantized to finite precision.

2.4 Virtual Character Representation

The use of MoCap technology for synthesizing complex human motion with virtual creation emerges as an effective solution towards generating realistic avatar's animation for both humanoid and articulated figures. In the current context, virtual character content is widely utilized and widespread across multiple areas such as entertainment, gaming and human computer interaction (HCI) applications due to the increasing demand of such realistic skeletal motions for different needs. The rapid advancement in processing capabilities and increasing functions of low-powered mobile devices have also brought upon a new evolution, where the application of VCA data is no longer limited to a standard powered machine or server system. With better graphics abilities and higher processing power packed within the lower powered devices, the use of realistic avatar's motion for interaction in virtual environment on laptops, PDAs and mobile devices is now feasible. The effective compression of VCA information plays an integral role in the application of such data on low-powered devices. Previously, techniques on 3D model compression focus on the mesh simplification and connectivity compression issues. These helped to achieve effective data reduction for mesh based

model as investigated in [25][99]. However, the use of such method may not be suitable for skeletal based 3D model where structural and temporal characteristics of human captured motions provide a greater free space for exploitation in achieving effective compression of data.

The MPEG community [12] plays an integral role to the development for 3D virtual character research. MPEG-4 [11] was initially introduced to cope with 2D/3D natural and synthetic objects that were not described in previous MPEG standards. The MPEG4-SNHC (Synthetic Natural Hybrid Coding) which later evolved to MPEG4-AFX (Animation Framework EXtension) [13] defines the specifications to handle functionalities for the streaming of virtual character animation and enables the VC to coexist and synchronize with other media types within the MPEG-4 defined hybrid environment. The H-anim [8] was proposed concurrently by the Web3D consortium to define the virtual avatar within the VRML environment. The framework was specifically focused on the avatar-like representation for which unlike MPEG-4 SNHC, whose definition is more generic and applicable for common articulated 3D objects. Both the MPEG-4 AFX and H-anim frameworks support the interaction between VC bones movement and skin deformation.

The representation for a VCA can be classified into two main groups namely body and facial parameters to describe both the body motion and facial expression of a virtual avatar entity. Recently, [36] proposed a scheme suitable for exploiting the structural attributes of the VC body parameters to achieve efficient compression. The framework is based on the combination of two earlier proposed BAP-sparsing [37] and indexing [38] techniques. The sparsing algorithm drops

and modifies predictive frames based on a freeze-release operation on BAP parameters dependent on their structural position. The BAP-indexing proposed the use of a look-up table to index the indices of the VC motion matrix. The effects of power consumption on mobile devices in respect to the reduced data size was also compared with the MPEG-BAP standard. In [104], the predictive and DCT (discrete cosine transform)[17, 110] scheme for exploiting the temporal coherence within the motion capture data is explored. They implemented the MPEG-4 BBA encoder and reported their findings on motion capture data. The BBA encoding concept introduced based itself on 4 key mechanisms for compression namely: prediction, frequency transform, quantization and entropy encoding. The MPEG-4 BBA reported a compression efficiency of up to 70:1 for virtual character animation. In [26], another approach of achieving compression for the motion database is presented. Here, correlations among motions within the database are used to achieve overall data reduction. The proposed lossy compression scheme first breaks down the motion capture database into short clips of motion sequences. The clips are represented using Bezier curves and processed using Clustered Principal Component Analysis (CPCA) [126][127] for dimensionality reduction. They reported a compression ratio performance of 1.1 to 35 for varying database of motion clips. [88] proposed the segmentation of human motion sequence into subsequence to enable poses within the segmented clips to be lying in low dimensional linear space [89]. The segmented clips are then compressed using Principal Component Analysis(PCA) [22] and Spline interpolation [34] is applied for further reduction in data size. The paper reported a compression ratio of up to 62 for varying motion sequences.

[58] discussed the adaption of human motions representation into MPEG-4 [11] and MPEG-21 [10] architectures. The paper focuses on the areas of virtual human representation adaptation based on the skeleton body based animation and their deformation facial based animation. The parametrization of DOF rotation for VCA is commonly done using either with Euler angles or Quaternion[29, 125]. [59] discussed the practical implementation of using exponential map for the representation of rotation and compared its advantages and limitations against other conventional schemes. Lastly, [43] proposed the progressive transmission of human motion sequences and its delivery across lossy communication network. The algorithm uses the structural attributes of the VCA data to achieve compression and scalability of virtual human animation. Recently, there is an increasing trend in the use of image processing standards on geometrical data. In [60], the static 3D model is encoded using images to reduce the dimension complexity of processing geometrical models and achieve compression using standard image tools. In [87], the applications for streaming of such image-based encoded 3D models were further presented.

2.5 Progressive transmission

The other challenge of 3D for practical application lies in the use of suitable channel protection scheme to ensure the efficient transmission of 3D information across erratic networks. Previously, unequal error protection schemes for image and wireless broadcast video content were proposed by Lei [35] and Wang [141] respectively. Lei [35] presented the findings that code rates in optimal allocation

are non-decreasing in nature and can be easily predicted using the last transmitted packet. A heuristic optimization suitable for real-time implementation was proposed in this paper. Wang [141] applied the H.264/SVC standard [119] to achieve scalable encoding of video streams. The UEP (Unequal Error Protection) scheme is applied to the different enhancement layers to achieve efficient transmission of higher impact layers. The source and channel rates were then optimized to achieve maximization of bandwidth utilization during transmission.

Error resilient encoding for 3D mesh was investigated in [98] to prevent error propagation during transmission. Park et al proposed the use of a shape adaptive data partition scheme to separate the 3D model into different segments before progressive encoding them using [97]. Their technique is suitable for 3D mesh data which are locally smooth and the partitions are highly correlated with its neighbor. Point-based which forms the other representation [78, 73, 93] which is by nature loss resilient towards transmission, has the limitation of having large data size for the efficient representation of all vertex points. The research in rendering and streaming of such representation were discussed in [115, 95, 82, 111].

Bici et al [31] and Al-Regib et al [21, 20, 18] considered the direct deliveries of 3D models using mesh based transmission taking into account both source and channel rates. Multiple description coding (MDC) [30][32][121] forms the other class of error transmission scheme used for error resilient transmission aside UEP. In the MDC scheme, descriptions are generated from the sub-meshes of a single model, each containing the full connectivity information to increase the decodability of the model. Aside from the conventional schemes, Mondet et al [94, 41]

proposed the progressive representations for streaming plant type models using retransmission scheme for lost packets. The accurate modeling of realistic plant model in virtual environment was discussed in [33, 50, 105, 91]. The 3D models transmission for virtual environment were discussed in [114][118][101]. View dependent and LOD strategies were explored and implemented in both [114][101] to improve the rendering time for large scale content delivery. In [39, 40], Cheng et al proposed a progressive transmission scheme based on an analytical model to investigate the progressive reconstruction of meshes across lossy channel. Gu and Ooi [63] discussed the problem of streaming progressive meshes across lossy network. Using PM [71] as the 3D representation, they introduced a graph-theoretic solution to optimize the dependency between packets during transmission. Li et al [85] proposed a novel concept of generic middleware for handling different type of triangle based progressively compressed 3D models suited for efficient network delivery of 3D progressive mesh models. Yang et al [145] proposed an optimized scheme to jointly consider the texture effects and mesh representation based on rate-distortion surface to facilitate progressive transmission of 3D model together with its respective texture information.

Lastly, [66] investigated the problem of transmitting 3D content across network from the transport protocol perspective. Based on the characteristic of data structure used to store the 3D representation, they proposed the design of On-Demand Graphic Transport Protocol (OGP) to address 3D transport protocol challenges such as node packetization and decision for retransmission. More recently, [19] introduced the use of an application layer protocol they named 3TP (3-D Model Transport Protocol) to intelligently decide the transmission of

information using either TCP or UDP to minimize end to end delays. Based on the use of 2-state Markovian model to simulate channel packet loss, their performance in term of R-D were presented and end to end delays were shown to outperform the approach of using purely TCP for transmission of such content.

2.6 Conclusion

In this Chapter, the related research and background studies for 3D models representation, compression and transmission techniques for both offline and streaming applications are reviewed. The Chapter has also discussed the emerging studies of image based encoding techniques for 3D static models and efficient representations for realistic human motion using Motion Capture data. The indepth discussion for a novel image based scheme for 3D representation named Spectral Geometry Image will be covered in the next chapter.

Chapter 3

Spectral Geometry Images (SGI)

3.1 Introduction

The revolution of 3D-TV technology brings experiences of users' interaction with their conventional image, video and 3D graphical contents to a new dimension. The successful showing of digital movies such as Avatar [3] demonstrated the huge demand for 3D visualization across diversified viewers. Recently, research had been undertaken to address the challenges faced for the representation, storage and delivery for such content. The state-of-art reviews in 3D video representation and storage can be found in [139].

Conventionally, existing broadcasting and streaming methodologies and applications [120][122] were solely focused for the mainstream natural videos and images. With the emerges of more 3D content, there is now an urgent need to address the better form of representation and more efficient delivery of such content across the public internet. The broadcasting and distribution of stereoscopic images to static and mobile users was presented in [75]. [147] discussed

the challenges faced for automatic conversion of 2D to 3D videos. [142] introduces a new quality measurement called Color plus depth level-of-details in 3D tele-immersive video and measures the perception thresholds of LOD within 3D videos. [124] presented a compression methodology for 3D videos based on concept of view-dependent rendering. A pre-rendering process for the real-time video is first performed omitting the occluded and invisible objects in the scene. The rendered images which consume much lesser channel bandwidth can now be transmitted directly to mobile devices for rendering. Recently, the concept of using graphics rendering contexts in real-time 3D video coding is introduced in [123], demonstrating its suitability in cloud gaming scenario.

There is a distinct difference between 3D video content and 3D graphical models with the latter having the attribute of interactivity. The deployment of mobile broadcasting technology [68] such as Digital Video Broadcasting-Handheld (DVB-H), Digital Multimedia Broadcasting (DMB) and MediaFLO supports the delivery of multimedia contents across mobile and portable devices such as cell phones and PDAs. [42] uses the MPEG-4 system to synthesize audio, 3D graphics, video, image and text content into a single content stream for interactive multimedia playback. However, although the MPEG-4 Binary format for Scene(BIFs) [1] which is supported in T-DMB provides a general representation for 3D model, there are limitations for the progressive delivery of 3D graphics across the broadcasting network due to the complexity and data representation in 3D.

3.1.1 Geometry Image and Spectral Analysis

Conventionally, 3D models are represented using polygonal or triangular meshes [23][48][71][96]. However, there are still limitations: (1) dependency between layers (2) the need for additional channel protection for mesh connectivity information, for such representation during the encoding and transmission process. Geometry image [60][87] is an effective solution for compressing shapes that are parameterized to the regular domain, like the rectangle or sphere to overcome the problem of connectivity preservation. However, since GI first re-sample the 3D geometry to the rectangular domain before performing the discrete wavelet transform (DWT), it is highly dependent on good parameterization and re-sampling processes to ensure the 3d models are well defined before compression. Spectral analysis [130][138][146] of 3D content's surface provides a solution for better 'understanding' of geometrical information. Karni and Gotsman presented the spectral coding algorithm [77] where the 3D model is first partitioned into several sub-mesh and finally, quantizes the spectral coefficients to finite precision. Our spectral geometry image approach is different from [77] for which we first partition the frequency domain into hierarchical layers, i.e., the base layer contains the smoothest geometry and the top layer contains the high frequency geometric details, and compress each layer with different compression rates. Furthermore, by taking advantage of the regular structure of geometry image, we can apply more advanced compression techniques (like JPEG2K) than the simple quantization used in [77]. Thus, our approach is more flexible and can achieve better compression ratio with less artifacts.

3.1.2 3D content delivery

The other challenge for 3D content delivery lies in the use of suitable channel protection scheme to ensure the efficient transmission of 3D information across erratic networks. Previously, unequal error protection scheme such as Bici et al [31] and Al-Regib [21] considered the direct delivery of 3D models using mesh based transmission taking into account both optimization of both source and channel rates. Mondet et al [94] proposed the progressive representations for streaming plant type models using retransmission scheme for lost packets. Since the 3D content are encoded using mesh representation, there is a need to ensure that the connectivity information for the entire mesh are well preserved using techniques of lossless transmission, retransmission or additional bits for channel protection. In addition, since the different layers of the mesh based model are often sequentially encoded, the earlier layers which contained the coarse information of the 3D model need to be correctly transmitted and decoded before the next layer can be used to reconstruct the more detailed model.

3.1.3 Contributions

Our work differs from the mentioned literatures, we proposed a new concept spectral geometry image(SGI) and developed a framework to deliver progressively encoded 3D models using image approach suitable for progressive transmission purposes. We combined the use of spectral analysis with the state of art Geometry Image(GI) to encode static 3D models into SGI for robust 3D shape representation. SGI separated the geometrical image into low and high frequency layers based on the 3D model's surface characteristics to achieve effective Level of

Details(LOD) modeling. Our experiment demonstrates that in term of flexibility and data size representation, the proposed SGI is more powerful compared to the conventional geometry image. In order to address the issue of transmission resilience for the delivery of SGI encoded images across lossy channels. An optimized joint source and channel coding framework is proposed to achieve optimal allocation of both source and channel bits.

3.1.4 Focus and Organization of Chapter

The focus of this chapter is to present quantitative analysis of the image-based encoding scheme, Spectral Geometry Coding for the encoding and transmission of static 3D models across packet loss networks. The simulation result will show that the proposed framework is effective in ensuring the smooth degradation of progressive 3D models across varying channel bandwidths and packet loss conditions. This chapter is organized as follows. The review for the image encoding scheme for the SGI using JPEG2000 [133] is presented in Section 3.2. In Section 3.3, the algorithm to construct the spectral geometry image is detailed. Section 3.4 discusses the image compression scheme. Next, Section 3.5 describes the challenges in data transmission across lossy channels and Section 3.6 presents the use of Joint Source and Channel method for the SGI representation to address such problems. The experimental results of the proposed system are presented in Section 3.7. Section 3.8 discusses the comparison of the SGI against current state-of-art. Finally, conclusions are given in Section 3.9.

Figure 3.1: 2-level discrete wavelet transform and decomposition for image Lena into subbands.

3.2 Review of JPEG2000

The JPEG2000(J2K) is the recent international image standard originally introduced to replace the successful JPEG. Unlike its predecessor, the J2K uses the concept of discrete wavelet transform(DWT) to decompose an image tile into wavelet sub-bands of LL(Low-Low), HL(High-Low), LH(Low-High) and HH(High-High) resolutions. Figure 3.1 demonstrates the example of an image decomposition into subbands using the two-level DWT.

The DWT is capable of achieving both lossless and lossy compression using either the reversible or non-reversible mode of compression. The reversible compression applies the LeGall (5,3) filter for lossless coding and the Daubechies (9,7) filter is adopted for higher and lossy mode of compression. A uniform and scalar quantization is applied to quantize the coefficients of the subbands $a_\beta(i, j)$, where β is denoted as the designated subband layer, into $q_\beta(i, j)$. The quantization is stated as follows:

$$q_\beta(i, j) = \text{sign}(a_\beta(i, j)) \frac{|a_\beta(i, j)|}{\Delta_\beta} \quad (\text{Eq. 3.1})$$

Δ_β denotes the quantization step size. For $\Delta_\beta = 1$, the reversible wavelet transform is used. The code-blocks with typical size of 64×64 formed from the quantization process are compressed using entropy encoding techniques for further data reduction. The J2K image encoding technique is a good solution for our SGI compression due to its flexibility and error resilience capabilities. The compression efficiency of the J2K is higher in comparison to the conventional JPEG scheme with the trade-off of having an increase to the processing complexity due to the DWT computation. The experimental comparison of the JPEG2000 against the existing state-of-art technique is presented later in Sect. 3.4.

3.3 Geometry Encoding

The following sections present the algorithmic details of constructing spectral geometry image.

3.3.1 Manifold harmonics transformation

This subsection briefly reviews the algorithm to compute the spectrum of Laplacian, i.e., manifold harmonics basis. More details can be found in [138]. Given a surface M represented by a triangular mesh $M = (V, E, F)$ where V , E , and F are the vertex, edge and face sets. The symmetric Laplace-Beltrami operators are defined as:

$$\begin{cases} \Delta_{ij} = 0 & \text{if } \{v_i, v_j\} \notin E \\ \Delta_{ij} = \frac{\cot \alpha + \cot \beta}{\sqrt{A_i A_j}} & \text{if } \{v_i, v_j\} \in E \\ \Delta_{ii} = -\sum_j \Delta_{ij} \end{cases} \quad (\text{Eq. 3.2})$$

where A_i and A_j are the areas of the two triangles that share the edge $\{v_i, v_j\}$ and α and β are the two angles opposite to that edge. Figure 3.2 shows the

Figure 3.2: Mesh diagram showing Laplace Beltrami operators.

visual representation for the Laplace-Beltrami operators. The eigenfunctions and eigenvalues of the Laplace-Beltrami operator are all the pairs (H^k, λ_k) that satisfy:

$$\Delta H^k = \lambda_k H^k \quad (\text{Eq. 3.3})$$

Since the Laplacian Δ is a symmetric matrix, its eigenvalues are real and eigenfunctions are orthogonal. The eigenvalues are sorted in an increasing order, $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. These eigenfunctions are the basis functions and any scalar function defined on M can be projected onto them.

For each vertex v_i , define a hat function $\Phi_i : M \rightarrow \mathbb{R}$ such that $\Phi_i(v_i) = 1$ and $\Phi_i(v_j) = 0$ for all $j \neq i$. Then, the geometry of M is represented by functions $x = \sum_i x_i \Phi_i$ (resp. y, z), where x_i denotes the x -coordinate of v_i . The eigenfunctions H^k can be represented as $H^k = \sum_i H_i^k \Phi_i$. Projecting the function x to manifold harmonics basis, we have

$$\tilde{x}_k = \langle x, H^k \rangle = \sum_i x_i H_i^k \quad (\text{Eq. 3.4})$$

Figure 3.3: Spectral analysis on 3D surface. The eigenfunctions of Laplace-Beltrami operator are orthogonal and serve the manifold harmonics basis. Row 1: the color indicates the function value. Row 2: the texture mapping shows the isocurves of the basis functions.

Figure 3.4: Reconstructing the 3D mesh from the frequency domain. The number above each model is the number of eigenfunctions used in surface reconstruction.

$\tilde{x}_k$ is the coefficient of k -th frequency of function x . Similarly, $\tilde{y}_k$ and $\tilde{z}_k$ for the functions y and z can be computed respectively.

To reconstruct the shape from the frequency domain, the coordinates of the vertex v_i are given by:

$$x_i = \sum_{k=1}^m \tilde{x}_k H_i^k, y_i = \sum_{k=1}^m \tilde{y}_k H_i^k, z_i = \sum_{k=1}^m \tilde{z}_k H_i^k \quad (\text{Eq. 3.5})$$

where m is the user-specified number of eigenfunctions. With the increasing number of eigenfunctions, the shape can be faithfully reconstructed from the frequency domain. Figures 3.3 and 3.4 illustrate the manifold harmonics transformation on the Bimba model.

3.3.2 Conformal parameterizations of 3D models to rectangular domain

A key step in constructing (spectral) geometry image is to parameterize the 3D model M to a rectangular domain $D \in \mathbb{R}^2$. Although there are many surface parameterizations techniques, the conformal parameterization is preferred due to its shape preserving property and numerical stability [62]. For the SGI, the focus is on the genus-0 closed surface. The example for the conformal parametrization of the genus-0 Bimba model is provided in Figure 3.5.

Topological modification: The topology is first modified by two cuts, i.e., one at the top and the other at the bottom of M . Let M' denote the resultant open surface. Note that M' has the same geometry of M , but is a topological cylinder.

Computing the uniform flat metric: Let g denote the Riemannian metric of M' . The idea is to compute a metric that is conformal to g and flat everywhere inside M' and the geodesic curvature is constant on the boundary $\partial M'$. Such a metric

Figure 3.5: Conformal parameterization of the genus-0 Bimba model. (a) The topology is first modified by two cuts, one at the top and the other at the bottom of the model. The cut surface M' is a topological cylinder. (b) The uniform flat metric by discrete Ricci flow is computed and M' is embed to a topological annulus. (c) Next, the topological annulus is mapped to a canonical annulus by a Möbius transformation. (d) The canonical annulus is cut by a line passing through the origin and conformally mapped to a rectangle. (e)-(f) The checker-board texture mapping illustrates the conformality of the parameterization.

is called uniform flat metric. It is proven that if the total geodesic curvature on each boundary is given, such a uniform flat metric exists and is unique.

In the implementation, the discrete Ricci flow [61] is used to compute the uniform flat metric. The targeted Gaussian curvature of each interior point is set to zero, i.e., it is completely flat, $\bar{K} = 0$, $v \notin \partial M'$. M' has two boundaries, $\partial M' = C_0 \cup C_1$, where C_0 is the boundary with the longer length. Then the total geodesic curvature of the boundary C_0 and C_1 is set to be 2π and -2π respectively, i.e., $\int_{C_0} \bar{k} = 2\pi$ and $\int_{C_1} \bar{k} = -2\pi$. It can be easily verified that the

Figure 3.6: Spectral geometry image. (a)-(b) show the geometry image of the Bimba model. (c)-(f) show the 3-layer spectral geometry image. The normal maps in (b) and (d) highlight the difference between SGI_1 and GI . To better view the high frequency layers (e) SGI_2 and (f) SGI_3 , the pixel values are normalized to $[0, 255]$.

total geodesic and Gaussian curvatures satisfy the Gauss-Bonnet theorem,

$$\int_{M'} K + \int_{\partial M'} k = \int_{M'} \bar{K} + \int_{\partial M'} \bar{k} = 2\pi\chi \quad (\text{Eq. 3.6})$$

where $\chi = 0$ is the Euler number of M' . It is proven that discrete Ricci flow converges exponentially fast [45] and the steady state is the desired uniform flat metric. With the uniform flat metric, the Gaussian curvature of interior vertices are zero, thus, the faces can be flattened one by one on the plane.

Conformal map to a canonical annulus: Note that the embedded surface $\phi(M')$ may not be the canonical annulus. Let $\phi(C_1) : |z - c_1| = r_1$ and $\phi(C_2) : |z - c_2| = r_2$ be the outer and inner circles of the topological annulus. We want to find a Möbius transformation $w : \mathbb{C} \rightarrow \mathbb{C}$ to map $\phi(C_1)$ and $\phi(C_2)$ to concentric circles with center at the origin. A Möbius transformation is uniquely determined by three pairs of distinct vertices $z_i \in \mathbb{C}$ and $w_i \in \mathbb{C}$, $i = 1, 2, 3$, such that $w(z_i) = w_i$. Set $w_1 = 0$ and $w_2 = \infty$, i.e., w_1 and w_2 are symmetric w.r.t. the canonical annulus. Therefore, the pre-images z_1 and z_2 are symmetric w.r.t. $\phi(C_1)$ and $\phi(C_2)$, i.e., $|z_1 - c_1||z_2 - c_1| = r_1^2$ and $|z_1 - c_2||z_2 - c_2| = r_2^2$. $z_3 = c_1 + (r_1, 0)$ and

$|w_3| = 1$ is also set, i.e., the radius of the outer circle in the canonical annulus is one. Then, the Möbius transformation is given by:

$$w(z) = \rho e^{i\theta} \frac{z - z_1}{z - z_2} \quad (\text{Eq. 3.7})$$

where $\rho = \frac{|z_3 - z_2|}{|z_3 - z_1|}$ and θ is an arbitrary angle.

Conformal map to a rectangular domain: The canonical annulus is cut by a line passing through $(0,0)$ and $(1,0)$. Finally, the cut annulus is conformally mapped to a rectangular domain by:

$$\psi(z) = |z| + i \arg z, \quad z \in \mathbb{C} \quad (\text{Eq. 3.8})$$

Putting them all together, the conformal parameterization $f : M' \rightarrow D \in \mathbb{R}^2$ is given by the composite map, which is guaranteed to be conformal (angle-preserving) and diffeomorphism.

$$f = \psi \circ w \circ \phi \quad (\text{Eq. 3.9})$$

where $\circ$ denotes function composition. The checkerboard texture mapping illustrates the conformality of the parameterizations.

3.3.3 Construction of spectral geometry image

Spectral geometry image is more flexible than the conventional geometry image due to its capability to separate geometry image into low- and high-frequency layers. Note that the low-frequency layers represent the rough shape and the high-frequency layers represent the detailed geometry. In the proposed framework, the user specifies the number of desired layers l and the reconstruction tolerance for each layer ϵ_i , $i = 1, \dots, l-1$. M_i denotes the reconstructed mesh

with m_i eigenfunctions where m_i is determined by finding the smallest integer such that:

$$\begin{aligned} & \arg \max_{\mathbf{k}} \max_j \sqrt{\sum_{k=m_i+1}^{\infty} (\tilde{x}_k H_j^k)^2 + (\tilde{y}_k H_j^k)^2 + (\tilde{z}_k H_j^k)^2} \\ &= \|M_i - M\|_{\infty}, \text{subject to } \epsilon_i \end{aligned} \quad (\text{Eq. 3.10})$$

The spectral geometry images are defined as follow:

$$\begin{aligned} & SGI_1 : M_1 \rightarrow D \\ & SGI_2 : M_2 - M_1 \rightarrow D \\ & \dots \\ & SGI_i : M_i - M_{i-1} \rightarrow D \\ & \dots \\ & SGI_l : M - M_{l-1} \rightarrow D \end{aligned} \quad (\text{Eq. 3.11})$$

Intuitively speaking, M_1 represents the coarsest reconstruction of the 3D model. The remaining layers M_i encode the displacement between the following two consecutive layers M_i and M_{i-1} and present the model with increasing quality until the original model is decoded in the top layer M_l . Thus, to reconstruct the geometry with the user-specified tolerance ϵ_i , the layers are simply added up to i . Figure 3.6 shows the 3-layer spectral geometry image of the Bimba model with the tolerances $\epsilon_1 = 0.04$ and $\epsilon_2 = 0.01$. The model is normalized to a unit cube.

Figure 3.7: JPEG2K vs JPEG-XR. One can clearly see that JPEG2K is more effective than JPEG-XR at low bpp , while JPEG-XR is more preferred for lossy compression of high quality results. Due to the smooth nature, geometry images have better measured PSNR compared to natural images from the experiment.

3.4 Spectral geometry image compression

In the previous section, the different layers of spectral geometry image from the 3D model are defined. Now, with low and high frequency geometry represented in images, existing image processing techniques to achieve compression and delivery of the model can be used. The JPEG2K [133] and JPEG-XR [52] are widely adopted standards due to their robustness and efficacy in compressing natural images. Figure 3.7 presents the PSNR performances of JPEG2K and the conventional JPEG-XR based on both natural images and geometry images.

It can be seen that JPEG2K is more effective than JPEG-XR to compress both kinds of images for moderate quality results, while JPEG-XR is more preferred for lossy compression of high quality results. Since JPEG2K adopted the

discrete wavelet transform for energy packing, it results in better PSNR for a given bit size compared to JPEG-XR. For the thesis, JPEG2K is adopted as the compression algorithm for both spectral geometry image and geometry image.

3.5 Channel coding

The Reed-Solomon (RS) code is used as the error control code to provide the SGI encoded image with some measure of reliability during the transmission across lossy communication channel. Typically, it can be described as an (n,k,t) code. The code block length n is limited by $n \leq 2^m - 1$ where m is the numbers of bits per symbol. The codeblock contains $k = n - 2t$ source symbols defined by the Galois fields over $GF(2^8)$. RS is a well known class of block codes among FEC that provides good erasure correction properties and is commonly used in storage devices for the correction of burst errors. Here, an (n,k,t) codes will be able to correct up to $2t$ (protection symbol) or t errors where each error is taken as two erasures.

3.5.1 Channel Model

The network plays a crucial role in the reconstructed quality of the 3D model upon transmission. However, the process of determining the individual packet behavior over a real-time channel is often complex in nature. In this work, the existence of a channel estimator based on two-state Markovian Gilbert-Elliot model [54] is adopted to simulate such characteristic. State 0 is denoted as a packet being correctly received and state 1 indicates a lost packet. $P(1|0)$ and $P(0|1)$ fully described the transitional probabilities between the two states.

3.5.2 Problem formulation

In this section, the allocation problem for both source and channel bits given the input of varying bandwidth limitations and channel loss characteristics is discussed. For the purpose of progressive transmission, the input spectral geometry image is decomposed into l image layers to support the reconstruction of partial bit-stream from coarse to high quality 3D models at the decoder. However, due to the dependency between the layers of such multi-resolution model, the effects of channel error on the decoded 3D model can be extremely significant during transmission. Thus, there is a need to exert some form of error control to ensure a measure of reliability is maintained in the presence of such errors. The forward error code aims to protect data against channel errors through the introduction of parity codes. In a conventional transmission system where bandwidth is limited, there is often a dilemma in resource allocation between the source and channel blocks where better quality encoding needs to be balanced against having sufficient channel protection. In this thesis, the joint source-channel coding system for SGI representation is considered. The proposed scheme mitigates the problem of resource allocation for both source and channel bits to achieve an overall improved 3D model quality with SGI upon transmission.

3.6 JSCC for SGI

In the earlier sections, SGI is introduced as an alternative 3D representation for progressive compression of complex 3D models. However, there is a need to propose an equally efficient channel coding scheme as preprocessor to ensure that the data are well protected during transmission. Interdependencies of layers

for progressive 3D model are often critical, thus packet loss during transmission could have a large impact on the final decoded quality of the 3D model.

In this section, the joint source and channel coding framework is presented to optimize the allocation of source bit for SGI and channel bits for Forward Error Correction (FEC). FEC is commonly used in digital communication for error corrections through the introduction of redundancies into the original bitstream. Generally, it can be classified into two main groups: Equal Error Protection (EEP) and Unequal Error Protection (UEP). EEP is defined as the equal allocation of FEC across the bitstream regardless of importance in contributions between the different layers. In contrast, UEP takes the contribution of different layers into consideration to unequally assign FEC to the different part of the bitstream to ensure that the more important layers are better protected. In our framework, the source encoding is handled by SGI. Since the 3D models are encoded as images and sorted into multiple layers of unequal contributions. The UEP technique is adopted during the joint optimization process to ensure that the more important layers are better protected during transmission.

Another unique characteristic of the SGI in comparison to mesh based model is the embedding of connectivity information within the SGI image itself. Mesh model often requires both positional data (V_x, V_y, V_z) and connectivity information for decoding. During transmission, the loss of connectivity information will render the unsuccessful construction of any mesh based model thus this data is often assumed to be transmitted in a lossless format. SGI solved this problem through the parametrization process, thus there isn't any additional bits or pro-

cess required to protect the connectivity information of the original 3D model. The UEP allocation for the FEC of the SGI is described as follows:

Let the set of channel code rate for the individual SGI layers be denoted as $C = \{C_1, C_2, \dots, C_{l-1}\}$ where $C_1 \geq C_2 \geq \dots \geq C_{l-1}$. The channel rate follows the pattern of non-decreasing to take into consideration importance of the different SGI layers. That is, more channel information in the form of RS code will be assigned to the higher priority lower layers as compared to the higher layers which contain the finer details of the model. The total bit budget, Q_{budget} defines the maximum bandwidth available during the transmission. Given an overall coding rate of R_{s+c} for both SGI source and RS channel codes, the objective is to optimally allocate number of bits such that the total distortion D_{s+c} is minimized.

$$\min D_{s+c}(r_s, r_c), \text{ subject to } R_{s+c} \leq Q_{budget} \quad (\text{Eq. 3.12})$$

r_s and r_c are denoted as the source and channel rates for the different SGI layers respectively. Previously, Cao [35] applied the technique of UEP for the progressive transmission of image across lossy channels. The concept was extended to scalable video across wireless network in [141]. More recently, Tian [134] presented the use of chunk based transmission, based on UEP of multiple objects within a 3D scene to achieve error resilient. Al-Regib et al applied the BOP (Block of Packets) on CPM bitstream during packetization and uses the UEP across the BOP to achieve efficient protection during progressive transmission of 3D models.

Figure 3.8: Packetization scheme. Figure demonstrates the packetization scheme and UEP allocation adopted for the SGI encoded model from the initial EEP configuration.

In our work, the JSCC technique is applied on the different SGI layers to optimize the allocation between channel and source codes. The packetization scheme for SGI is based on concepts from [141] but extended for SGI transmission of 3D models. The illustration of the packetization scheme is presented in Fig. 3.8. The width and height layered j are denoted as w_i^j and h_i^j respectively. j represents the layer number where $j=1,2,\dots,l-1$. i denoted the numbers of iterations during optimization. The packet number and packet size are denoted by N and M respectively. During the optimization, w_i^j and h_i^j are iteratively adjusted to customize the allocation of FEC to the different layers of SGI based on their importance. In order to prevent the loss of consecutive packets containing correlated information when burst errors occur, the allocation of FEC is done horizontally and the packets are vertically packed.

The probability of recovering the i layer is denoted as $P_i \in i = 1, \dots, l-1$. Let $\lambda = D_1 P_1$, λ shows the initial distortion obtained from decoding the base layer upon transmission. The resultant distortion $D_{s+c}(k)$ obtained from the decoding

of remaining layers, can now be easily determined. The differential improvement $\varepsilon_{i,k}$ is shown in the following equation:

$$\varepsilon_{i,k} = \sum_{i=1}^{N-w} D_{i,k}(P_{i,k|C_k}) \quad (\text{Eq. 3.13})$$

Here, $D_{s+c}(k) = \lambda + \varepsilon_{i,k}\xi_k$, if $1 \leq k \leq l-1$ and $D_{s+c}(k) = \lambda + \varepsilon_{i,k-1}\xi_{k-1} + D_k\xi_k$, if $k = l$, where ξ_k denotes the decodability of additional SGI layers. For the computation of the packet loss probability, $P(m, n)$, let n be the total number of packets transmitted across the channel and m is the number of packets lost during the process. For layered i , where RS codes is applied for packet error protection. $P_i = \sum_{m=0}^{N-w} P(m, n)$ denotes the probability that i th SGI layer is decodable. For experimental purpose, the earlier discussed two-state Markov model is used to simulate the lossy nature of the erratic channel.

3.6.1 Distortion metric

To measure the quality of the compressed surface M' , d_H [77] calculated is based on the average of norm of geometric and Laplacian differences is used here to determine the model's quality.

$$d_H(M, M') = \frac{1}{2} \left(d_r + \sqrt{\frac{1}{V_n} \sum_{i=1}^{V_n} \|L(v_i) - L(v'_i)\|^2} \right) \quad (\text{Eq. 3.14})$$

V_n is the number of vertices in M and M' . Note that $L(v_i)$ denote the geometric Laplacian for vertex v_i . d_H measures the smoothness of the compressed mesh through the calculation of Laplacian operator and is able to reflect the visual

Figure 3.9: Conformal parameterization of the test models.

quality better compared to Euclidean measurement such as Root Mean Square (RMS) which only considers the geometric distance.

Figure 3.9 shows the test models and their parameterizations and Fig. 3.10 compares the performance of GI and SGI of the Bunny model. Note that both GI and SGI discard high frequency details at high compression ratio. The low-frequency layer of SGI is more smooth than that of GI and thus lead to less artifacts. Fig. 3.11 compares the performance of GI and SGI of the Bunny and Foot model. When reconstructing the geometry from multi-layer SGI of different resolutions, the bottom layers are up-sampled to the resolution of the top-most layers. This upsampling usually smoothes the low-frequency geometry, but it does not change the top-most layer that contains the high-frequency details within the user-specified range. Since the Laplacian operator encodes the differential coordinates [77] representing the local details, it is insensitive towards the small-scale

Figure 3.10: Mean curvature error d_H measures the visual quality. Row 1: the GI of resolution 256×256 ; Row 2: the 3-layer SGI of resolution 64×64 , 128×128 and 256×256 . The numbers below each figure are the *bits per pixel* (bpp) and the mean curvature error d_H . SGI has smaller d_H at low bpp because the high frequency layer is discarded and the low-frequency layer has less distortion than GI.

deformation of the bottom layers. As a result, the SGI-64/128/256 outperforms the GI-256 in terms of mean curvature error. In this thesis, the visual metric d_H is used to determine the contribution from each source layer of the compressed model. Although other form of Euclidean error measurements such as Hausdorff distance or root mean square rms can be used, d_H which measure the smoothness of the surface of a model, can better reflect the visual quality of a compressed model in term of error.

 (a) mean curvature error (Bunny) d_H

 (b) mean curvature error (Foot) d_H

Figure 3.11: Shape compression using geometry image (GI) and spectral geometry image (SGI). GI is of resolution 256×256 for both models. The two SGIs with 3 layers are constructed respectively. Both Bunny and Foot models are of resolution 64×64 , 128×128 , and 256×256 . As shown in (a) and (b), SGI's performance is better than GI for varying compression ratio.

3.7 Simulation Results

A distortion optimal rate allocation algorithm is proposed to provide the SGI encoded model with efficient channel protection during transmission. In the

Figure 3.12: Comparison of error protection schemes. (a)-(b) show the mean curvature error for both (a)EEP and (b)UEP scheme with fixed source rate of 0.8bpp, 0.1-0.4bpp channel rate for the Bunny model. (c)-(d) show the mean curvature error for both (c)optimized EEP and (d)JSCC scheme with overall bit rate of 0.8-1.2bpp for the Bunny model. Horizontal axis denotes the packet lost rate P_{LR} in (%). The mean curvature error d_H is represented by $10\log_{10}(1/d_H)$.

experiment, the bitstream over packet loss channel is simulated using a two-state Markovian model as the channel estimator. The average block length B_L is set as 9.57 and the channel loss rate was determined using Gaussian distribution over 100 cycles for 2%, 5%, 12%, 20% and 28% to simulate the randomness in a broadcast channel.

In consideration for the source coding rate of the 3D model, a subset of $Q_r = (5, 10, \dots, 300)$ is selected, where Q_r denote the compression ratio, for the i layered of SGI image encode using J2K standard to ease the computation in the

Figure 3.13: Comparison of error protection schemes. (a)-(b) show the mean curvature error for both (a)EEP and (b)UEP scheme with fixed source rate of 1.2bpp, 0.1-0.4bpp channel rate for the Foot model. (c)-(d) show the mean curvature error for both (c)optimized EEP and (d)JSCC scheme with overall bit rate of 1.2-1.6bpp for the Foot model. Horizontal axis denotes the packet lost rate P_{LR} in (%). The mean curvature error d_H is represented by $10\log_{10}(1/d_H)$.

simulation. The Bunny (SGI-64/128/256) and Foot (SGI-64/128/256) are used to show the experimental results. The source bit rate and overall bit rate are 0.8bpp and 0.8-1.2bpp for the Bunny model, 1.2bpp and 1.2-1.6bpp for the Foot model in both EEP and UEP setting. The performance (d_H) over the range of average packet error loss rate P_{LR} are illustrated in Fig. 3.12 and Fig. 3.13.

For each experiment, the results from the proposed JSCC algorithm are compared against the performances of the following schemes. Fig. 3.12(a) and Fig.

3.13(a) depict the results for the EEP method where EEP assigns equal channel rates to each SGI encoded layers respectively without considering the importance of the layer. The simulation results using the UEP method are presented in Fig. 3.12(b) and Fig. 3.13(b). For UEP, unequal channel rates are allocated to the SGI layers based on their layer importance. Since the SGI is constructed based on the frequency components of the 3D model, the lower layers which contain the rough shape of the original model carries the more essential information to be protected during transmission. Fig. 3.12(c) and Fig. 3.13(c) show the results from the optimized EEP scheme. The overall bit rates are 1.2bpp and 1.6bpp for the bunny and Foot model. Unlike the earlier EEP scheme where the source bit rate is fixed, the optimized EEP method determined the best source and channel bit rate set and assigned channel bits equally across all layers.

It is important to note that our proposed scheme differs from the conventional transmission methods. For SGI, the 3D static model is first encoded into multiple images of difference importance. Unlike conventional methods for 3D model transmission, the connectivity information of the 3D model is implicitly encoded within the SGI, thus there isn't a need for additional protection to ensure the connectivity information lossless delivery. Secondly, the source and channel bits of the SGI layers are jointly optimized before transmission. In Fig. 3.12 and Fig. 3.13, the proposed solution showed that it is able to provide a more graceful degradation of decoded model quality compared to the optimized EEP method and consistently shows a better decoded model quality against the other solutions across varying packet loss rate. From our observation, the proposed JSCC method outperforms the EEP, UEP and optimized EEP by a d_H of 0.678, 0.411

and 0.493 respectively for the overall bit rate of 1.6 bpp and (P_{LR}) of 28% for foot model. d_H denotes the mean curvature error represented by $10\log_{10}(1/d_H)$. For the bunny model, the proposed JSCC method also shows an improvement of 0.1-0.48 d_H across the different methods. In the second experiment, the proposed *SGI* scheme is compared against the state-of-art *GI* technique applying similar JSCC technique to both methods for a fair comparison. Due to the efficient application of JSCC scheme for both SGI and GI method, both schemes show a smooth decrease in quality (d_H) during channel simulation across varying channel condition. Fig. 3.14 shows that the SGI is capable of ensuring a better quality 3D model decoded throughout the simulation. This is due to the efficient allocation of the lower frequencies geometries of the SGI into the initial layers of the SGI encoded image. During the channel allocation, more channel bits will be allocated to the lower layers compared to the GI scheme, thus better resilience of the earlier layers can be ensured. Fig. 3.14 demonstrate the improvement of SGI over GI across varying channel condition.

3.8 Discussion and comparison against state-of-art techniques for 3D representation

In this section, we discuss the comparison of spectral geometry image against state-of-art techniques in 3D representation and show its limitations.

Comparison against PM: Polygonal mesh modeling is by the far the most common 3D representation format (.wrl, .ply, .3ds, etc) used in the world today. Each 3D object is composed of a collection of key attributes which include: Vertex, Edge and Face information. A vertex can be define as a single point or

(a)

(b)

Figure 3.14: Comparison of d_H between JSCC allocation scheme for both GI and SGI method. (a)-(b) show the simulation results for (a)Bunny and (b)Foot model across varying packet loss rate. GI and SGI are both of resolution 256×256 . As shown, SGI outperforms GI for various packet loss condition. The number below the figure are the packet lost rate (%) and the vertical axis is the d_H denoted in $10\log_{10}(1/d_r)$.

coordinate of the 3D object in the three dimensional space. For any two vertices that can be commonly connected, this can be described as an edge. Face consists of the required vertices or edges information needed to define a polygon or triangle on a 3D surface. Hoppe has introduced the use of progressive meshes for representation of 3D objects and inspire the area of research for progressive compression using mesh based modeling. Progressive mesh (PM) is an effective solution to deliver 3D models across error prone networks since its hierarchical structure facilitates reducing latency time for previewing large-scale, complex models. However, the quality of PM based model is highly dependent on the correct decoding of the connectivity information, and previous LOD layers of a 3D mesh is often necessary for progressive reconstruction of the model. In the context of SGI, the delivery order of the SGI layers is additive in nature thus making it suitable for transmission across unreliable channel. In addition, the connectivity information of the 3D model is also implicitly encoded in the image, thus additional protection for such information is not required for SGI. The comparison against mesh based model is summarised as follow:

Progressive Mesh(PM)

- Hierarchical structure reducing latency time to preview model.
- Data consists of both position coordinates and connectivity information.
- Connectivity information crucial for model reconstruction, additional resource or lossless transmission required to protect the data.

Spectral Geometry Meshes(SGI)

- 3D model is decomposed into different layers suited for LOD transmission.
- Reduction in dimensionality as 3D model is now encoded as image.
- Connectivity information is implicitly embedded within image due to parameterization, no added resource required to protect the data.
- SGI layers are additive in nature.

Comparison against Geometry Image: The major difference between the proposed spectral geometry image and geometry image is that the SGI performs the manifold harmonics transform (MHT) on the 3D model and then partitions the frequency domain with the user-specified tolerance. Finally, the partitioned layers are re-sampled to the 2D domain by the parameterization. Thus, the spectral analysis of SGI is independent of the parameterization. The GI re-samples the 3D geometry to the rectangle domain, and then performs the discrete wavelet transform (DWT) on the 2D rectangular grid. Thus, compared to SGI, GI's spectral analysis highly depends on the parameterization and re-sampling. As shown in Fig 3.15, a parameterization of poor quality may result in the jaggedness artifacts due to the high anisotropy. However, SGI with the same parameterization shows more robust results.

Comparison against Bandlets: Bandlets extend the wavelets to capture the anisotropic regularity of edge structures, thus, they are very promising in processing images with rich geometric structures [92]. Peyré and Mallat applied bandlet to compress geometrically regular objects, like geometry images and

normal maps, and showed that bandlets improve the wavelets on complex 3D models [100]. Our approach makes use of JPEG2K (which in turn is based on wavelets) to compress spectral geometry images. Bandlets are more effective than wavelets to approximate smooth edges and sharp features. However, there is no gain to apply bandlets to the highly smooth geometry. Within spectral geometry image framework, only the high-frequency layer encodes the edges and sharp features. Thus, it would be promising to combine spectral geometry images with bandlets such that the low- and high-frequency layers are compressed by wavelets (JPEG2K) and bandlets separately.

One limitation of the proposed scheme lies in the surface parameterization. For the SGI scheme, the genus-0 model is parameterized to a rectangular domain using conformal parameterization [62]. The conformal parameterization minimizes the angle distortion, however, usually results in large area distortions. For example, the Bunny ears and Gargoyle wings and head have large area distortion as shown in Fig. 3.9. For such models, a polycube [70, 140] is a possible parametric domain solution for the geometry image and can be further investigated in future works .

3.9 Conclusions

In this Chapter, an image based method for 3D representation suitable for adoption in conventional broadcasting standard is proposed. The problem of delivering progressively encoded 3D content across packet erasure channels is investigated. The Chapter discussed the implementation of a novel 3D encoding method, Spectral Geometry Image(SGI), which is more robust, and efficient in

comparison to the state of art Geometry Image technique for 3D models representation. It was shown that by coupling SGI together with the proposed JSCC allocation scheme, an effective framework for delivery of progressively encoded 3D model is realized. In our framework, the source encoding is handled by SGI. Since the 3D models are encoded as images and sorted into multiple layers of unequal contributions. The UEP technique is adopted during the joint optimization process to ensure that the more important layers are better protected during transmission. The experimental results show that the proposed method outperforms the conventional GI in term of coding efficiency and error resilience performance simulated for varying packet lost rates.

(a) DWT applied to 2D regular grid
with parameterization of good quality

(b) DWT applied to 2D regular grid
with parameterization of poor quality

(c) MHT applied to 3D mesh

(d) re-sampling of MHT results
by the parameterization of poor quality

Figure 3.15: Discrete wavelet transform is highly dependent on the parameterization and re-sampling. (a) and (b) show DWT applied to 2D regular grid with different parameterizations. The middle and right figures show the reconstructed meshes with LL subband and LL+HL+LH subbands respectively. Clearly, the parameterization with poor quality results in the jaggedness artifacts due to high anisotropy. (c) Manifold harmonics transform is performed on the 3D meshes directly, thus, independent of the parameterization. From left to right, the reconstructed meshes with the number of eigenfunctions, 160, 500 and 1000 respectively. (d) shows the re-sampling of MHT results by the parameterization of poor quality. It has much less artifacts than that of (b).

Chapter 4

LOD and compression of Virtual Character Animation

4.1 Introduction

The increase in demand for better quality 3D models and graphical animations is prominent with the technology advancement in processing capabilities and ease of graphic creations for the last few decades. With the recent surges in both the gaming and entertainment industries in alignment to the evolving Internet towards a content based sharing ideology, the cause for finding better 3D representations, faster content creation techniques, more efficient compression schemes and better delivery of such graphical contents dynamically across different users becomes critical. Among the different 3D generation schemes, the use of MoCap technology for synthesizing complex human motion with virtual creation emerges as an effective solution towards generating realistic avatar's animation for both humanoid and articulated figures.

4.1.1 Virtual human representation

The research in the areas for VCA is fairly new in comparison to work done for 3D mesh representation and compression. Skeletal based 3D model has the unique characteristics of both structural and temporal coherency which causes the applications of 3D mesh compression techniques unsuitable, if it is directly applied on the VCA. Previously, [36] proposed the Model based power aware method to achieve efficient compression of the VCA. The structural attributes of the VCA are exploited through the combination of BAP sparsing and indexing techniques. [37] drops and modifies predictive frames based on freeze-release operations on the BAP based on their structural position. [38] uses the look-up table technique to index the indices of the DOF matrix. The proposed works achieve good compression efficiency. However, it does not address the suitability of the representation for progressive delivery. The parameterizations of the DOF rotations are commonly carried out using either using Euler or Quaternion techniques [29, 125] for their robustness and ease of rotation complexity. [59] discussed the practical implementation of using exponential map for the representation of rotation and compared its advantages and limitations against two other conventional schemes. In the Chapter, the Euler and Quaternion representation are independently reviewed to compare their effectiveness for adoption in both SVCA and VCAM schemes.

4.1.2 MPEG community for VCA development

The MPEG-4 [5][6][16][103] community plays an important role in the development of VCA. Recently, the H-anim 1.1 specification [8] was set up to initiate a

standard framework for representing humanoids creation to ease the complexity in creating and usage of such content. However, with VCA being a relatively new area of research, several interesting directions such as an efficient representation, compression and delivery of such information across bandwidth limited channels are yet to be thoroughly studied. Another important aspect lies in the unique characteristic of the VCA data structure. The existing compression techniques for conventional 3D model do not consider the implicit coherence within the anatomical structure of a human skeletal model and lack portability for transmission consideration. This can cause an unacceptable quality degradation of the character animation during the transmission of such information across the network.

4.1.3 Contributions

In this Chapter, SVCA is introduced to achieve efficient LOD for the VCA. Unlike previous literatures, the SVCA method addresses the limitation for progressive reconstruction of VCA data. The proposed work first converts the Mocap representation into DOF matrix and uses the bit-plane encoding scheme which is common in scalable video compression to achieve effective LOD of skeletal based animation. Next, the data representation is further optimized by exploiting the structural behavior of the VCA node and re-sequence the data packets based on their structural contribution. This ensures the VCA data is scalable in nature to dynamic bandwidth constraints and providing optimized decoded output at any designated bit-rates. Next, VCAM introduces a new VCA representation using image based approach to further improve the compression efficiency of VCA data.

VCAM first mapped the DOFs matrix from the mocap information as an image map. Since temporal coherency are presented within the joints sequences of the VCA, they are further exploited to improve compression efficiency.

4.1.4 Focus and Organization of Chapter

The focus of this Chapter is to propose two different VCA compression schemes that make use of (1) Structural coherence in the anatomical structure of the VC and (2) Motion correlations within the VCA to achieve efficient compression. The experimental analysis of the two methods will be shown and further discussed. This chapter is organized as follows. Section 4.2 introduces the matrix representation of the VCA motion data. Sections 4.3 and 4.4 provide an overview for the comparison of Euler and Quaternion representations. Sections 4.6 and 4.7 discuss the framework and algorithm of the proposed Scalable Virtual Character Animation (SVCA) and Virtual Character Animation Mapping (VCAM). Section 4.8 details the experimental results and comparison between the proposed methods. Lastly, the conclusion is provided in Section 4.9.

4.2 VCA matrix representation

This section introduces the matrix representation used to generalize the VCA format. Firstly, the motion deviation within the joints of the skeletal avatar across the frame sequences of the VCA can be described using a collection of rotation and translation movement defined by μ_i^j . Here, i denotes the number of degree of freedom (DOF) within the skeletal model where $i = 1, 2, \dots, n$ and j indicates the frame number in the animation sequence for $j = 1, 2, \dots, m$. $\mu_i^j, \forall i, \forall j$ describe the

Figure 4.1: Skeletal representation of the VCA model denoting the hierarchical characteristic between individual joints.

Figure 4.2: VCA of run and leap sequence 49.05 and running sequence 02.03. The motion sequence (a) and (b) consist of a total of 164 and 173 frames respectively. The original AMC files for (a) is approximately 128KB and (b) 135KB in size.

entire animation. The rows of the matrix from $(\mu_i^1, \mu_i^2, \dots, \mu_i^m)$ contain the joint modifications of a single DOF across the frames and $(\mu_1^j, \mu_2^j, \dots, \mu_n^j)$ represent the DOFs of a single frame j .

For $1 \leq i \leq 3$, the initial 3 DOFs define the root positions of the VC model. The remaining DOFs of the skeletal model consist of the Euler angles used by

different joint nodes for obtaining the corresponding joint's position. The structure of joint hierarchical formation is shown in Fig. 4.1. The Virtual Character consists of a total of 31 joint nodes with 62 DOFs. The initial 3 DOFs denote the root position of the skeleton model and remaining $n-3$ DOFs of the VC consist of the Euler angles needed by the individual node for decoding the respective joint positions. For consistency, $n=62$ is applied for all skeletal model in this thesis. However, it is worth noting that the proposed scheme is still applicable for alternative models with a different n value. Fig. 4.3 summarizes the representation for each joints id and its corresponding DOF limitation. The left column consists of the id for different joints part. The row entity shows the corresponding motion modification in the respective axis. The numerical value indicates the sequence number of the DOF in the motion data. In this context, the motion deviation within joints of the skeletal avatar across frame sequence of the VCA can be represented using a collection of rotation and translation data.

4.3 Euler Angles Representation

Euler angle is a common form of representation to define the angular displacement of a rigid body. It was named after the mathematician Leonhard Euler for his work in using angular parameters to define the rotation of an object in coordinates convention. Euler angles have several advantages in the sense of their practicality for real-world application and fast computation to enable its considerations for future streaming modules. Euler uses only three rotational parameters to define the orientation of an object which is one of the smallest and most optimize way to represent an orientation. In the following example, three rotational angles

Joint (ID)	Tx	Ty	Tz	Rx	Ry	Rz	
R (0)	1	2	3	4	5	6	Root
1				X	X	X	LHIPJOINT
2				56	57	58	LFEMUR
3				59			LTIBIA
4				60		61	LFOOT
5				62			LTOE
6				X	X	X	RHIPJOINT
7				49	50	51	RFEMUR
8				52			RTIBIA
9				53		54	RFOOT
10				55			RTOE
11				7	8	9	LOWERBACK
12				10	11	12	UPPERBACK
13				13	14	15	THORAX
14				16	17	18	LOWERNECK
15				19	20	21	UPPERNECK
16				22	23	24	HEAD
17					37	38	LCLAVICLE
18				39	40	41	LHUMERUS
19				42			LRADIUS
20					43		LWRIST
21				44		45	LHAND
22				46			LFINGER
23				47		48	LTHUMB
24					25	26	RCLAVICLE
25				27	28	29	RHUMERUS
26				30			RRADIUS
27					31		RWRIST
28				32		33	RHAND
29				34			RFINGER
30				35		36	RTHUMB

Figure 4.3: VC table to summarize the representation for each joints id and its corresponding DOF limitation. Initial 3 DOF denotes the position of the root of the skeleton. Remaining i-3 DOF of the VC consist of the Euler angles needed to specify each joint position.

denoted by γ, β, α are used to describe the arbitrary rotation of a rigid body in the 3D Euclidean space. We also demonstrate the use of Euler angle for the rotational of a 3D coordinate point denoted by (x, y, z) to its rotated position of (X, Y, Z) . The conventional Z-Y-Z order for coordinate system, rotations in

counter clockwise (right-hand rule) are shown below. The rotation matrix M_R is denoted as follows:

$$M_R = R_z(\alpha)R_y(\beta)R_z(\gamma) \quad (\text{Eq. 4.1})$$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 4.2})$$

The first rotation uses the Euler angle γ to rotate the initial coordinate position (x,y,z) to (x',y',z') around the original z axis.

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \quad (\text{Eq. 4.3})$$

The second rotation uses the Euler angle β to rotate the earlier coordinate position of (x',y',z') to (x'',y'',z'') around the y' axis.

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 4.4})$$

The last rotation uses the Euler angle α to rotate the coordinate position of (x'',y'',z'') to the final displacement of (X,Y,Z) around the z'' axis. Figure 4.4 presented the respective rotations given the three Euler angles. The rotation matrix M_R is defined by a sequence of rotations which does not commute. This implies that the sequence of rotation around the individual axes impact the result of the final orientation. M_R is denoted as a general 3×3 rotation matrix as follows:

Figure 4.4: Euler rotation using Z-Y-Z.

$$M_R = \begin{bmatrix} \cos \alpha \cos \beta \cos \gamma - \sin \alpha \sin \gamma & \sin \alpha \cos \beta \cos \gamma + \sin \gamma \cos \alpha & -\sin \beta \cos \alpha \\ -\sin \alpha \cos \beta \cos \gamma - \sin \gamma \cos \alpha & -\sin \alpha \sin \gamma \cos \beta + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \\ \sin \beta \cos \gamma & \sin \beta \sin \gamma & \cos \beta \end{bmatrix} \quad (\text{Eq. 4.5})$$

$$M_R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (\text{Eq. 4.6})$$

The use of Euler angles provides the advantage of fast computation, less storage requirement and easy interfacing for animators to perform rotation due to its intuitiveness in representation. However, Euler does have its limitations in aliasing problem arising from independency between rotations and the occurrence of Gimbal Locking which limits the Euler's application for interpolation. The use of Quaternion provides an alternative solution to overcome the 'Gimbal Lock' characteristic in Euler at the expense of greater storage requirement.

4.4 Quaternion

The concept on the use of Quaternion was invented by Sir William Rowan Hamilton [64, 65] in 1843. The key motivation behind the creation was to determine

the complex number representation for the three-dimensional space which will enables the multiplication corresponding to the scaling and rotation in a 2D plane. The Quaternion encompasses two main components of Scalar and the Imaginary portion which holds the 3D vectors components. Its basic notation can be described as follows:

$$Q = \gamma + xi + yj + zk \quad (\text{Eq. 4.7})$$

γ is defined as the scalar component where $\gamma \in \mathbb{R}$ and $\beta \in \mathbb{R}^3$ for the vector portion. Extending the complex number to a 3D space, let $i^2 = j^2 = k^2 = -1$ where $ij = k$, $jk = i$, $ki = j$ and $ji = -k$, $kj = -i$, $ik = -j$.

$$Q = [\gamma, (x, y, z)] = [\gamma, \beta] \quad (\text{Eq. 4.8})$$

Quaternion [67, 83] was common in computer animation applications due to its flexibility and robustness to handle smooth interpolation. In the following, a basic review on the mathematics operation for better Quaternion understanding is presented.

Quaternion Magnitude The calculation for the magnitude of the Quaternion set $Q = (\gamma, \beta)$ is denoted as follows:

$$\|Q\| = \|\gamma, \beta\| = \|\gamma, (x, y, z)\| = \sqrt{\gamma^2 + x^2 + y^2 + z^2} \quad (\text{Eq. 4.9})$$

The magnitude comprises both the scalar and vector components similar to the magnitude calculation of 2D complex numbers.

Quaternion Addition In the addition for two Quaternion Sets, the individual scalar and vector components of set where $Q = (\gamma, \beta)$ and $Q' = (\gamma', \beta')$ are independently added.

$$\begin{aligned} Q + Q' &= (\gamma + ix + jy + kz) + (\gamma' + ix' + jy' + kz') \\ &= (\gamma + \gamma', i(x + x') + j(y + y') + k(z + z')) \quad (\text{Eq. 4.10}) \end{aligned}$$

The second set of Quaternion is defined as Q' where $Q' = (\gamma', \beta') = (\gamma' + ix' + jy' + kz')$.

Quaternion Multiplication The multiplication process involve the product of individual components governed by the mathematical distributive law.

$$\begin{aligned} QQ' &= (\gamma, \beta) * (\gamma', \beta') \\ &= (\gamma + ix + jy + kz) * (\gamma' + ix' + jy' + kz') \\ &= (\gamma\gamma' - \beta \cdot \beta', \beta \times \beta' + \gamma\beta' + \gamma'\beta) \quad (\text{Eq. 4.11}) \end{aligned}$$

The multiplication of two Quaternion Sets, ie: $QQ' \neq Q'Q$ are non-commutative in nature. The $\times$ denotes the cross product in three dimensional space.

Quaternion Rotation The use of rotation matrix is possible for animation sequence to enable interpolation [117] and derivation for optimize Key-framing processing. The conversion of the Quaternion set to the rotation matrix can be easily defined using the following description.

$$Q(\gamma, (x, y, z)) = \begin{bmatrix} \gamma^2 + x^2 - y^2 - z^2 & 2xy - 2\gamma z & 2xz + 2\gamma y \\ 2xy + 2\gamma z & \gamma^2 - x^2 + y^2 - z^2 & 2yz - 2\gamma x \\ 2xz - 2\gamma y & 2yz + 2\gamma x & \gamma^2 - x^2 - y^2 + z^2 \end{bmatrix} \quad (\text{Eq. 4.12})$$

Quaternion has the significant advantage over Euler angle in applications where there is a need to process the interpolation between orientation. The key trade-off involve the one additional parameter required to define the full Quaternion representation, which translate to a respective increase in storage size and processing time for animation sequences involving large angular displacement. Quaternion in nature is more difficult to work with and is not as intuitive in comparison to the Euler Angles.

4.5 Bit-plane Coding

The use of bit-plane(BP) coding technique on DCT coefficients to achieve fine granularity scalability(FGS) for the MPEG-4 video standard is well known and demonstrated in [86]. FGS presents the concept of representing a single video sequence using both base layer and enhancement layers. It offers an effective solution to provide flexibility for video bit-stream data to be partially decodable at any bit rate to achieve optimized quality. This is often critical in bandwidth limiting environments such as mobile applications devices and communication. Previously, several other techniques [7][9] were compared against the bit-plane coder for MPEG-4 FGS implementation. BP coding was eventually selected as the standard for its ease in implementation and comparable coding efficiency against other state-of-arts. For SVCA presented in Section 4.6, conventional transform coding (DCT) is applied to achieve better energy compaction of DOFs since individual DOF sequences from VCA often exhibit strong coherency across time. BP coding technique is applied to achieve scalability for the VCA bit-stream.

Figure 4.5: Bit-plane Coding technique example.

An example for BP coding scheme is presented in Fig. 4.5 for better understanding on the algorithm from [86]. The original data presents the information obtained from the quantized DCT coefficients and has completed the encoding process from run-length coder.

Since the largest absolute value of the data block is 18, thus a total of 5 bit-planes will be required to fully represent the data. From the bit-plane information, the binary data are then sorted to form the (RUN,EOP) symbols. RUN denotes the number of consecutive zeros before a 1 and EOP denotes if there are any more 1s in the bit-plane. The (RUN,EOP) enables further compression of the bitstream using variable length coding technique.

In previous sections, reviews for the VCA conventional representations and BP coding technique were presented as background for better understanding. The data representation for the VCA and implementation for the SCVA algorithm is described in the following Sections.

4.6 Scalable Virtual Character Animation

The joints of the skeletal avatar animation is independently defined as a series of angle modifications $(\mu_i^1, \mu_i^2, \dots, \mu_i^m)$ respective to the joint node and treated as a sequence of discrete signals. The data exhibits the characteristic of closeness in positional changes and strong coherence across a VCA in the absence of abrupt scene changes. In order to take advantage of such characteristics, transform coding techniques from [86] is applied to achieve better energy compaction for each DOF. The transform coefficient of DOF i is denoted by $(C_i^1, C_i^2, \dots, C_i^m)$. It is demonstrated that by coupling the transform technique with the signal characteristic of the motion avatar information, an efficient LOD scheme can be realized similarly for virtual character animation. We summarised the processes to encoded the VCA using BP- VCA as follows:

Generalized Algorithm for BP-VCA

- (1) Store DOF data from VCA into Matrix (m x n).
- (2) Apply DCT to generate $C_i^1, C_i^2, \dots, C_i^m$ and normalize each row of DOF.
- (3) Apply bit-plane coding technique and sort the layers by $L_0, L_1, \dots, L_{enh_lay}$.

The $\max, \min(C_i^1, C_i^2, \dots, C_i^m)$ of each DOF are used to normalize the individual coefficients to the desired bit-planes resolution. The generated bit-planes are then assigned into L_0, L_1, L_{enh_lay} where L_0 defines the base layer of the VCA and remaining are the enhancement layers. Each enhancement layer contains a resolution of 2 bit-planes incremental with data size of $(m \times n \times 2)$ bits. Figure 4.6 and Fig. 4.7 demonstrate the joint effects of the transform process and bit-plane

Figure 4.6: Reconstruction of DOF-50 with progressive increment of bit-plane 1-4.

coding scheme on the VCA signal. The method is titled BP-VCA (bit-plane VCA) for experimental comparison in the later chapter. The initial bit-planes of the DOF coefficients contain the more vital information in describing the general motion behavior of the VC. This translates to having a better representation of the entire VCA without decoding the full information of the VCA as shown in Fig. 4.6 and Fig. 4.7.

Next, the unique hierarchical structural of the virtual character is considered to address the problem of sequentially transmitting the different layers of the avatar's DOF information. The BP-VCA highlighted in previous paragraphs, provides the technique to exploit the advantage of coherency within the VCA motion to achieve energy compaction and LOD of the VCA. However, the as-

Figure 4.7: Reconstruction of DOF-50 with progressive increment of bit-plane 4-16 layers.

sumption for different DOFs having equal importance and priority was made and the DOF information is transmitted in accordance to their initial indexing. This is not an optimized solution as the joints and nodes of the VCA have a unique parents/child relationship with one another. Thus, the order for transmitting different DOF plays a crucial role in achieving the optimize quality of VCA a given bit-rate. To address the mentioned problem, a fast heuristical method is discussed next to optimize the bit-planes transmission sequence to achieve a good quality motion reconstruction for a given bit rate. The quality improvement of the VCA per packet received is defined as $\Delta DEF = DEF(k) - DEF(k + 1)$ where k is the current packet. The objective is to ensure the $\min \Delta DEF$ to enable optimization in the packet transmission sequence.

Displacement error per frame (DEF) is originally introduced by [36] to calculate the sum of errors for all joints and normalising it by the total number of frames. In the bit-plane VCA algorithm, ΔDEF denotes the improvement in VCA relative motion quality during optimization to determine the next best packet for transmission. More details of the DEF metric is provided in the later Section 4.7.3.

The greedy-based distortion reduction algorithm is introduced to rearrange the order of transmission for each packet and stored its resulting sequence list. The selection of DOF elements for packet k is defined as E_k . E_k is composed of different bitplanes and DOF components of the encoded VCA. During each iteration, the $\min DEF = DEF(E_k, E_{k+1})$ is determined. The optimization ensures the selection of the next enhancement bit-stream introduce the least possible error distortion to the current VCA stream. The new sequence of transmission generated from the process is kept within a new index listing table for use during transmission. We summarized the process for SVCA as follows:

Generalized Algorithm for SVCA

- (1) Store DOF data from VCA into Matrix (m x n).
- (2) Apply DCT to generate $C_i^1, C_i^2, \dots, C_i^m$ and normalize each row of DOF.
- (3) Apply bit-plane coding technique.
- (4) Sort bit-plane increment as packets.
- (5) Optimize sequence of bit-plane improvement w.r.t $\Delta DEF = DEF(k) - DEF(k+1)$.

Figure 4.8: Comparison of DEF vs data size between BP-VCA and SVCA for run and leap VCAs sequence.

(6) Store new index list for DOF.

Figure 4.8 demonstrates the differences in performance between the earlier discussed BP-VCA and the optimized SVCA approach. For the BP-VCA, it is noticeable that although the bit-stream of the avatar's motion is transmitted progressively, the quality improvement of each packet varies greatly across time. The result demonstrates the impact on the DEF value due to the hierarchical characteristic of the avatar's model. Since different DOF of the model contribute unequally to the decoded avatar's motion, the early transmission of the more important DOF through sequence indexing can result in an overall improved avatar's motion in a shorter time. Observation from Fig.4.8 shows that the

proposed SVCA ensures a smooth continuous drop in DEF compared to the initial progressive encoder without the sequence indexing. The SVCA scheme exploits the structural behavior of the individual nodes and re-sequences the packets according to their structural importance and contribution.

4.7 Virtual Character Animation Mapping

Section 4.6 discussed the approach of using SVCA to achieve scalability of the VCA data. The comparison of DEF vs data size presented in Figure 4.7 demonstrated the performance of the progressive reconstruction of the VCA making it suitable for applications of progressive transmission of skeleton motion across dynamic bandwidth constraint. Although the proposed SVCA offers a good solution to construct LOD details for the VCA, the compression performance of the algorithm is not factored in the discussion. In this section, a new technique for VCA representation and compression is proposed. The proposed scheme makes use of the humanoid motion characteristic and achieves effective data reduction in the model's representation using conventional image processing technique. The implementation of the image based VCA is presented as follows.

In the case of a skeletal model animation, the joints modification $(\mu_1^1, \dots, \mu_i^j, \dots, \mu_n^m)$ is used to decode the final position of the skeletal joint. It can be observed that since each node of the avatar exhibit small deviation between frames of animation, that motion signal obtained from each DOF exhibit strong coherence in the temporal direction as demonstrated in Fig. 4.9. Matrix μ_i^j is partitioned into k image segments to further increase the image correlation before compression. This improves the overall encoding efficiency of the motion data. The (min,max)

Figure 4.9: For both images, the motion information is encoded as a JPEG2000 [133] image before compression where $m=62$, (a) demonstrates the image mapping for run and leap sequence and (b) shows the image mapping for cartwheeling sequence. Both image map exhibit the characteristic of strong temporal coherence along the row entity.

Figure 4.10: The sequence of image maps denote motion of the skeletal avatar for the walking animation. The motion data consist of 652 frames of motions and $m=62$ in this example. Individually, each mapping is normalized and compressed using JPEG 2000 encoder [14].

values of μ_i^j for $j=1,2,\dots,m$ of each DOF are next determined. Lastly, the motion data for individual joint is normalized and encoded as 16 bits grayscale image.

4.7.1 VCA decoded quality for varying motion frames length

It is important to note that the length of the motion sequence plays an important role in determining the quality of the decoded VCA. Given a longer sequence in comparison to a shorter one, the correlation within the sequence decreases comparatively resulting in poorer decoded VCA quality. To address this issue, the motion data μ_i^j is partitioned into k image segments of μ_n^n . Similar to the use of macroblocks in video compression encoding, the use of multiple image maps for the VCA yields a better compression efficiency and smaller degradation in decoded skeletal motion quality. The image maps created from the walking sequence are shown in Fig. 4.10.

4.7.2 Image map compression

JPEG2000 [133] is an image compression standard that uses the state of art wavelet technology to provide flexibility and improved compression performance over the conventional JPEG [108] standard. It was accepted as an international standard in December 2000 and was used in multiple 3D geometry based encoding literatures [60],[87]. The J2K for image compression of the VCA is adopted for its robustness and efficiency in compressing image dataset, and providing the flexibility of progressive encoding for future investigation.

In the work for SVCA and VCAM, the key objective is to achieve efficiency in both compression and scalability of VCA data. In order to mitigate processing time, the entire VCA sequence was partitioned into k images of μ_n^n as discussed in Section 4.7.1 to be separately encoded prior to any transmission. In this context, each image on its own contains the information necessary for the decoding of n

frames of VCA sequence at any instant as shown in fig. 4.10. This eliminates the need to apply JPEG2000 for each frame of VCA data thus reducing the complexity and processing time during encoding.

4.7.3 Error metric

In order to evaluate the quality of the decoded VCA model which undergoes compression in comparison to its original counterpart, there is a need for an objective error metric to measure the VCA differences. The displacement error per frame (DEF) proposed in [36] is used for this purpose. Let the decoded VCA joint positions from the VCA image map be denoted as $\gamma(J_{\beta j})$ where $\beta = [1, 2, \dots, n_j]$ and $j = [1, 2, \dots, m]$. n_j and m are the total joints and total frames of the VCA respectively. $J_{\beta j}$ is the position coordinates in $\mathbb{R}^3$ to define the individual joint position of the skeletal model across frame. The original VCA sequence without undergoing the compression process is used as the benchmark for comparison between the two models. This is defined as $O(J_{\beta j})$. The error function is presented as follow:

$$DEF = \frac{1}{m} \sum_{j=1}^m \sum_{\beta=1}^{n_j} \sqrt{(\gamma(J_{\beta j}) - O(J_{\beta j}))^2} \quad (\text{Eq. 4.13})$$

The error measurement compares the displacement error across frames between the joints position of the original VCA and decoded virtual character's motion which undergoes the image compression. The joint position give a better overview of the VCA quality compared to its DOF counterpart, thus giving a better on the subjective evaluation of the resulting animation quality. In the next section, the performance of the proposed scheme for different motion data

will be shown and the comparison of compression efficiency against the SVCA scheme will be presented.

4.8 Experimental Simulation

The Acclaim skeleton/motion capture file (ASF/AMC) is used as the standard format for the experiment. ASF/AMC file format is originally developed by Acclaim company for the storage of skeleton data captured from optical tracker. The format consists of two main files namely: ASF (Acclaim skeleton file) and AMC (Acclaim Motion Capture) which defines the base pose of the VC and subsequent motion data of the VC captured in the AMC. More information detailing the descriptions of the format can be found in [2]. It is worth noting that the proposed algorithm extends to other formats including MPEG-4 BAP which can be used as the common representation layer for the motion capture information.

Table 4.1 presents the compression simulation of 12 motion data for lossless compression in the image domain. In the experiment, the VCA mapping technique is able to achieve an average of 7.363 compression ratio against the original AMC file without significant degradation in the reconstructed motion quality outperforming the existing technique proposed in [36] reporting an average of 1.7-2.5 compression ratio for varying DEF of 0.17-14. The only quality lost in our process is due to the double to integer casting during the image mapping stage. However this lost in motion quality is insignificant with $DEF \leq 0.1$ for each sequence. Next, the simulation results for motion files 49_05 and 02_07

Table 4.1: Results for the compression performance of the VCA mapping technique presented. Here, FN - original motion capture filename, Desc - the description of the motion sequence, AMC - the AMC file size in (KB), Fr - number of frames in sequence, Cs - Compressed file size, CR - compression ratio of uncompressed to compressed motion files.

<i>FN</i>	<i>Desc</i>	<i>AMC(KB)</i>	<i>Fr</i>	<i>Cs(KB)</i>	<i>CR</i>
02_01	<i>walk</i>	267.0352	343	35.4902	7.5242
02_03	<i>run_jog</i>	135.0176	173	18.1631	7.4336
02_04	<i>jump_bal</i>	380.3223	483	51.5859	7.3726
02_05	<i>punch</i>	1449.2314	1854	189.6084	7.6433
02_07	<i>swordplay</i>	1749.8291	2251	230.3467	7.5965
02_10	<i>washself</i>	2056.2861	2645	278.5479	7.3822
49_01	<i>walk</i>	510.3652	652	71.0371	7.1845
49_05	<i>run_leap</i>	127.8887	164	17.0664	7.4936
49_06	<i>cartwheel</i>	375.8672	481	50.3770	7.4611
49_14	<i>dance</i>	484.0762	619	67.5723	7.1638
49_19	<i>balance</i>	748.9629	957	107.2979	6.9802
49_21	<i>acrobatic</i>	1888.5732	2422	265.1973	7.1214

Table 4.2: Results for the lossy compression of VCA mapping technique for sequence 49_05 is reported. Here, R - denotes the compression ratio used in JPEG 2000 standard, AMC - original motion data file size in (KB), Cs - the compressed file size of the motion file (KB), DEF - Displacement Error per frame, CR - compression ratio of uncompressed to compressed motion files.

<i>R</i>	<i>AMC(KB)</i>	<i>Cs(KB)</i>	<i>DEF</i>	<i>CR</i>
1	127.8887	17.0664	0.0139	7.4936
2	127.8887	9.8271	0.4504	13.0138
3	127.8887	6.5449	2.5651	19.5401
4	127.8887	4.8770	5.5150	26.2231
5	127.8887	3.8584	8.5333	33.1455
6	127.8887	3.3008	10.6463	38.7450

(where original amc file has 130,958 bytes and 1,791,825 bytes respectively) are shown.

In the experiment, the R (Compression ratio-J2K) for the image mapping is varied and the results obtained from the lossy compression are reported. The re-

Table 4.3: Results for the lossy compression of VCA mapping technique for sequence 02_07 is reported. Here, R - denotes the compression ratio used in JPEG 2000 standard, AMC - original motion data file size in (KB), Cs - the compressed file size of the motion file (KB), DEF - Displacement Error per frame, CR - compression ratio of uncompressed to compressed motion files.

R	$AMC(KB)$	$Cs(KB)$	DEF	CR
1	1749.8291	230.3467	0.0042	7.5965
2	1749.8291	135.0957	0.1203	12.9525
3	1749.8291	89.4092	0.7005	19.5710
4	1749.8291	66.8535	1.5276	26.1741
5	1749.8291	53.2998	2.2838	32.8299
6	1749.8291	45.0264	2.9547	38.8623

sulting compression ratio and their respective distortions in DEF are presented. Tables 4.2 and 4.3 present the simulation results from the lossy compression experiment. It can be observed that as the R of the JPEG2000 encoder is increased, the proposed scheme is able to achieve an effective compression ratio of up to 38 without much degradation in terms of DEF . Subjectively from observation, for a compression ratio ($R=4$) and below, it is difficult to observe the differences between the original motion compared to its compressed form. In Table 4.3, another interesting observation can be made. The resultant DEF results from the 02_07 swordplay sequences demonstrated a better DEF result compared to its counterpart 49_05 run leap sequence for similar compression ratio. The VCA sequence for swordplay consists of smaller deviations in motion as compared to other VCA sequence (run and leap). This results in greater coherency across time which enables a better compression performances being reported from the experiment.

The result from the proposed VCA mapping scheme is compared to the earlier

Table 4.4: The table shows the rate distortion comparison between the proposed scheme compared to its SVCA counterpart for the run and leap sequence. In the experiment, the results from the individual compression scheme is presented and compared. Here, Cs denotes the compressed file size of the motion file (KB), DEF - Displacement Error per frame and CR - compression ratio of uncompressed to compressed motion files.

<i>Method</i>	<i>Cs(KB)</i>	<i>DEF</i>	<i>CR</i>
<i>VCA – Map</i>	17.0664	0.0139	7.4936
<i>SVCA</i>	17.0600	0.1439	7.4963
<i>VCA – Map</i>	9.8271	0.4504	13.0138
<i>SVCA</i>	9.7990	1.6800	13.0510
<i>VCA – Map</i>	6.5449	2.5651	19.5401
<i>SVCA</i>	6.5190	11.9200	19.6178

introduced SVCA scheme. SVCA jointly considered the hierarchical structure of the VC model with its temporal coherence to achieve efficient compression. In the former method, a conventional transform technique is used instead of the wavelet scheme adopted by its counterpart. The entropy encoding scheme is also not applied in the earlier scheme. Table 4.4 presented the rate distortion comparison between the two techniques for similar bit rates setting in both case. From the table, it can be concluded that the VCA-mapping is able to achieve a greater compression ratio respective to the data size as compared to the SVCA scheme. Table 4.2 demonstrated that the VCA-mapping scheme is capable of compression predominance of up to 38:1 ratio of the original motion size. For the SVCA scheme, the maximum compression ratio is approximately 16:1 for a decoded VCA of reasonably good visual motion quality. If the compression ratio is further increased, the visual distortion on the VCA is no longer ignorable and will be distinctive to the human eyes.

4.9 Conclusion and Discussion

In this Chapter, the framework and implementation for both the Scalable Virtual Character Animation (SVCA) and Virtual Character Animation Mapping (VCAM) are shown. SVCA is proposed to provide scalability and compression of the virtual character animation through the use of conventional transform and bit-plane coding techniques. VCAM suggested the use of an image based approach towards the compression of VCA data. The experimental results showed that the VCAM method has competitive performance compared to the SVCA algorithm for rate-distortion performance.

Chapter 5

Clustering Approaches for Character Animation Representation

5.1 Introduction

The previous Chapter discussed the implementation and experimental analysis for both the SVCA and Virtual Character Animation Mapping techniques. Simulation results have shown that the two techniques are capable of compression ratio of up to 38:1 with tolerable visual distortion in the skeletal animation. Any further increase in compression ratio of the VCA will lead to a significant increase of the perceived visual distortion in the skeletal animation. In this Chapter, the VCAI (Virtual Character Animation Image) is introduced to overcome this limitation in performance.

5.1.1 VCA compression techniques

VCA sequence has the unique characteristic of having both temporal and structural coherency across time. Previously, several research has focused on either one of the characteristic to exploit the inherent redundancies to achieve better

compression performances. The compression for VCA data using structural information of virtual human information was demonstrated by [37]. Based on the standard compression and decompression pipeline for MPEG-4 BAP, an alternative stage called BAP sparsing was included to drop and modify predictive frames based on freeze-release operations on the BAP. Combined with a BAP-indexing algorithm for efficient indexing of DOFs information, the work presented an alternative approach for streaming VCA information across low-powered mobile devices. [104] has explored the concept of using predictive and DCT to exploit the temporal coherency of the VCA. The implemented BBA encoder performs prediction, frequency transform, quantization and entropy encoding to achieve efficient data compression. A compression ratio of up to 70:1 against the original VCA was reported from their experiment. Similarly in [26], the compression for a collection of motions in a database is investigated. The proposed method adopted the clustered principal component analysis for dimensionality reduction after the motion clips in the databases are represented using Bezier curves. A compression performances of up to 35:1 is reported from their experiments.

5.1.2 Contribution

Unlike previous works, VCAI uses the approach of image compression and clustering to jointly exploit the correlations between the structural and temporal data of the VCA to achieve efficient data compression. In addition, the VCAI uses a MMF (modified motion filter) to reduce the motion 'jittering' in VCA due to J2K [133] compression to further improve animation quality. For completeness, the experimental analysis of the two clustering schemes (K-mean and Fuzzy

C-Mean) will be presented. The VCAI outperforms the existing literatures and achieves a competitive compression ratio of up to 120:1 with minor visual artifacts on the VCA. The presentation for the proposed VCAI method across highly compressed range is available online at [15] for future references and comparisons.

5.1.3 Focus and Organization of Chapter

The focus of this Chapter is to propose the VCAI technique that jointly considers the structural coherence in the anatomical structure of the VC and motion correlations within the VCA to achieve efficient compression. The comparison of k-means and FCM clustering will be investigated. The experimental analysis of the two methods will be shown and further discussed. This chapter is organized as follows. Section 5.2 discusses the framework and algorithm of the proposed VCAI scheme. The experimental results of the proposed method are presented in Section 5.3. Lastly, the conclusion is provided in Section 5.4.

5.2 Proposed Framework for VCA representation

The MPEG-4 [11] body animation parameter (BAP) technique provides an efficient platform for the representation and compression of virtual human models. However, MPEG-4 BAP for virtual human sequences has yet to fully address the progressive compression for such data and coupling the problem with an absence to jointly exploit the structural and temporal coherency of the virtual human data during lossy compression, unacceptable quality degradation of the character animation can occurred during the transmission of such information across the network.

The VCA compression method discussed in this chapter seeks to address the mentioned problem for joint exploitation of both structural and temporal coherency within VC data. VCAI introduces a novel approach to encode the VCA data as images and jointly considers the structural and temporal coherency of the VCA using fuzzy clustering algorithm to achieve compression. In addition, since the VCA data are now encoded as images, the progressive reconstruction of the VCA is made possible. VCAI method is built with considerations to the existing MPEG-4 standard to allow its portability into the standard for future comparison. We discussed the algorithm and implementation of the VCAI in the following subsections.

5.2.1 Temporal coherence within motion

The joint modifications of the skeletal model use to decode the positions of the skeletal joint node are denoted as $(\mu_1^1, \dots, \mu_i^j, \dots, \mu_n^m)$. During the encoding process, the (min,max) values of μ_i^j for $j = 1, 2, \dots, m$ of each DOF are determined. The motion data for the different joints are then normalized and encoded into a 16 bits grayscale image suited for transmission. It can be observed that since each node of the avatar shows small deviations between frames of animation, the motion signal obtained from each DOF will exhibit strong coherence in the temporal direction as demonstrated in Fig. 5.1(a)-(b). Thus, image transform techniques can be applied to further exploit such characteristic and achieve better energy compaction during compression. Fig. 5.1 shows the VCAI encoding process with motion smoothening using 02.07 swordplay sequence as an example. Another important consideration lies in the length of the motion sequence.

Figure 5.1: VCAI encoding process with motion smoothening. (a) shows the original DOF trajectory values against frame number for 02_07 sequence. (b) shows the image mapping of DOF values to grayscale image. (c) denotes the VCAI which consider both anatomical structure of VC and motion coherence. (d) shows the VCAI with MMF for motion smoothening.

Given a longer sequence of frames in comparison to a shorter one, the correlation within the sequence decreases comparatively resulting in a lower decoded VCA quality. To address this issue, matrix μ_i^j is partitioned into k image segments of μ_n^n each to further increase the VCAI's correlation before compression. Similar to the use of macroblocks in video compression, the use of multiple image maps

for the VCA yields a better computation efficiency and smaller degradation in decoded skeletal motion quality. The discussion to this point explains the basis of current technique exploiting the temporal characteristic of VC's motion to achieve better data compression performance. However, a brief observation of Fig. 5.1(b) exposes the inherent dissimilarity of motion signals across DOFs when the individual motion modification parameters are positioned according to their original indexes. The following sections propose a solution to re-index the different motion data according to their structural positions and discuss the use of a clustering approaches to achieve clustering of similar DOFs parameter to form more efficient image maps before compression.

5.2.2 Clustering Approach

K-mean Clustering Approach Intuitively, clustering algorithms are commonly used to identify and group similar patterns together and segregate the ones that are different in characteristics into their respective clusters. It is also a common tool used in feature extraction [143] and image segmentation applications. This subsection briefly discusses the use of the k-mean algorithm to compute the index list based upon the similarity between DOFs information according to their hierarchical position on the VC. Since the skeletal avatar contains inherent data similarity due to the parent-child relationships between the joints of the Virtual Character model, this unique trait is used as an essential criterion for the clustering process.

The DOF motion can be denoted by μ_i^j where $i = 1, \dots, n$ defines the individual joint node and $j = 1, \dots, m$ represents the different frame. Each DOF μ_i can

be denoted an n - dimensional vector due to the partitioning of μ_i^j into k segments of μ_n^n for better correlation. Using k-mean algorithm, the n_x number of DOFs are clustered into k sets of $S_x = (S_1, S_2, \dots, S_k)$ where $k \leq n$. The centroid for the clusters is denoted by α_x . The sum of squared criterion is used here as the main function to determine if clustered objects are well separated. The function is defined as:

$$\epsilon_s = \sum_{x=1}^k \sum_{\mu_i \in S_x} \|\mu_i - \alpha_x\|^2 \quad (\text{Eq. 5.1})$$

During each iteration, the centroid α_x will be recalculated based on the current partitioning.

$$\alpha_x = \frac{1}{n_x} \sum_{\mu_i \in S_x} \mu_i \quad (\text{Eq. 5.2})$$

It should be noted that the DOFs data contains the Euler modifications of joints for 3D dimensional axis. Thus, to improve the performance of the k-mean, the initial DOF indexes are sorted based on their respective axis before clustering. Next, to record the changes in index order for the different DOFs, an order index list is maintained throughout the whole encoding process. The index list is made up of entries of 6 bits by n numerical values to constantly keep track of the changes in DOFs index orders, and is sent prior to the transmission process of the VCA images to the decoder for ease of decoding. The implementation of k-means clustering scheme for the exploitation of VCA data is simplistic in nature and achieves fast speed in computation. However, it also has the characteristics of non-deterministic and iterative, whereby the cluster size needs to be pre-determined and the algorithm is sensitive to the initial condition. The

Fuzzy C-mean(FCM) algorithm [144] introduced in the following paragraphs is an effective solution to resolve this problem.

Fuzzy C-mean clustering approach The full VCA sequence is earlier defined as $\mu_i^j, \forall i \forall j$. $(\mu_i^1, \mu_i^2, \dots, \mu_i^m)$ is considered for clustering. In FCM, each data point in this case μ_i^j , has some graded membership or degree of belonging to individual cluster. The graded coefficient $\psi_\beta(\mu_i^j)$ defines the measurement of membership respective to the cluster. The cluster membership to each data point is normalized as:

$$\sum_{\beta=1}^C \psi_\beta(\mu_i^j) = 1 \quad (\text{Eq. 5.3})$$

Unlike conventional k-means algorithm where each data point is assumed to be in exactly one cluster, the centroid in FCM algorithm is determined by the mean of all points and weighted by the graded membership to different clusters. The centroid is defined as:

$$\alpha_\beta = \frac{\sum_{\gamma=1}^n \psi_{\beta\gamma}(\mu_i^j)^M \mu_i^j}{\sum_{\gamma=1}^n \psi_{\beta\gamma}(\mu_i^j)^M}, 1 \leq M \leq \infty \quad (\text{Eq. 5.4})$$

M is any real number greater than 1 which influence the weightage. C is denoted as the number of clusters here. The solution to determine the graded membership is defined as:

$$\psi_{\beta\gamma}(\mu_i^j) = \sum_{\beta=1}^C \left(\frac{\|\mu_i^j - \alpha_\gamma\|}{\|\mu_i^j - \alpha_\beta\|} \right)^{\frac{2}{1-M}} \quad (\text{Eq. 5.5})$$

The optimization process constantly updates the graded membership $\psi_{\beta\gamma}(\mu_i^j)$ and clusters centers α_β during each iteration. The objective here is to minimize the cost function where

$$J_M = \sum_{\gamma=1}^n \sum_{\beta=1}^C \psi_{\beta\gamma}(\mu_i^j)^M \|\mu_i^j - \alpha_\beta\|^2 \quad (\text{Eq. 5.6})$$

During the optimization, the means of the data points eventually separate and the graded membership progressively shift towards the 0 and 1 values until the termination criterion is achieved. Further details of the FCM process can be found in [107]. Fig. 5.1(b) and Fig. 5.1(c) show the decoded VCA images from the FCM process. $\| * \|$ denotes the Euclidean norm used to measure the similarity of the data against the centroid. In [137], the Pearson correlation distance is considered instead of the Euclidean approach. However, since the computation cost of the algorithm is substantially higher and considering that the future use of VCA is inclined towards network transmission, the Euclidean distance is the preferred choice here.

5.2.3 VCAI compression and motion smoothening

The J2K standard is adopted for its performances and robustness in compressing both natural and geometry images as earlier studied in Section 3.4. J2K also offers better artifact performances over JPEG where ‘blocking’ and ‘ringing’ artifacts are both significantly prominent in JPEG for highly compressed image. For the case of J2K, since the encoding process is not constrained to the blocks encoding limitation, thus ‘blocking’ artifact is non-present here. The ‘ringing’ artifacts also known as Gibbs phenomenon, cause undesired signal oscillations

around the sharp transition within the signal. This effect occurs within both algorithms and translates to motion ‘jittering’ in the VCA. Here, a modified motion filter (MMF) based on Gaussian low pass filter is applied. The MMF preserves the main signal components in the VCA containing the fundamental motions and remove the high frequency noise components. For image compression, the Gaussian based filter is not an ideal solution for removing the ‘ringing’ artifacts in image as it blurs the image edges causing an undesired degradation in the image quality perceptually. On the contrary, observations from the experiments showed that such results are subjectively reversed for the VCAI animation. Technically, although the MMF blurs the VCAI in the image domain as shown in Fig. 5.1(d), this process translates to the smoothening of the motion trajectories in the VCA, thus causing the resulted perceptual quality of the animation to be significantly improved.

5.2.4 VCAI mapping and Error metric

The clustered VCA images earlier generated from Sect. 5.2.2 rearranged the DOFs positions according to their structural similarities, thus increases the spatial coherence in VCAI substantially. The VCAI is then compressed using J2K to generate the resultant 16 bits image prior transmission. Let $M_i^j, \forall i=1,2,\dots,n$ and $\forall j=1,2,\dots,m$ denote the compressed bitstream. m is the total frames in a VCA sequence and n is the total DOFs parameters here. Each set of compressed cluster VCA images produces a corresponding distortion measure (DEF) and data size Cs values dependent on the clustering process and channel conditions. Recall that the original VCA sequence is initially partitioned into k segments

before compression, the compressed data size of individual VCAI segments can now be defined as $C_1, C_2, \dots, C_k$ and the final compressed size from all J2K images is denoted as $Cs = \sum_{i=1}^k C_i$. Lastly, the J2K images are decoded back to $\mathbb{R}^3$ where joint positions denoted by $\gamma(J_x, J_y, J_z, \beta)$ can be used to quantify the decoded VCA quality. β is the joint index where $\beta = [1, 2, \dots, n_j]$ where n_j is the total number of joint nodes in the VCA. In this Chapter, the displacement error per frame (DEF) proposed in [36] and normalized distance from [104] are used as the criteria for objective quality evaluation.

For DEF, the resulting motion of all joints position from the decoded VCA model is denoted as γ where $\gamma(J_x, J_y, J_z)$, describes the 3D joint positions for the skeletal model across all frames. $O(J_x, J_y, J_z)$ is defined as the original VCA sequence without undergoing the compression process thus providing a benchmark for comparison between the two methods. The error function is described as follows:

$$DEF = \frac{1}{m} \sum_{j=1}^m \sum_{\beta=1}^{n_j} \sqrt{(\gamma_{\beta j}(J_x, J_y, J_z) - O_{\beta j}(J_x, J_y, J_z))^2} \quad (\text{Eq. 5.7})$$

The error measurement (Eq. 5.7) is based on the evaluation of joints position obtained from the original $O(J_x, J_y, J_z)$ and decoded $\gamma(J_x, J_y, J_z)$ virtual character's motion. The normalized distance metric from [104] provides an alternative objective measurement to determine the distortion. The error metric is based on the motion parameters $(rot_x, rot_y, rot_z, \dots)$, V' represents the decoded set of DOFs motion parameters and V^O is the original. The distortion metric is shown in (Eq. 5.8).

$$D_{norm} = \sqrt{\frac{\sum_{j=1}^m \sum_{i_m=1}^{n_mot} (V'_{i_m,j} - V^O_{i_m,j})^2}{m \times n_mot}} \quad (\text{Eq. 5.8})$$

In (Eq. 5.8), the index i_m is used to describe different motion parameters used to represent the rotation angles of the VCA. $i_m = [1, 2, \dots, n_mot]$ where n_mot denotes the total number of motion parameters used. From the experiment, it is observed that the joint positions give a better overview of the VCA quality, relating better to the subjective evaluation of the resultant animation quality compared to the normalized distance.

$$\delta(DEF, Cs) = \min_{DEF, Cs} (M_i^j) \quad (\text{Eq. 5.9})$$

Finally, $\delta(DEF, Cs)$ denotes the set of VCA images that results in the best compression efficiency and error performances determined from the iterative process (Eq. 5.9).

5.3 Experimental Result

This section presents the simulation performances for the VCAI method using VC motions with varying characteristics and complexity. In the experiment, the Acclaim skeleton (ASF) and motion capture file (AMC) were used as the standard formats for the experiment test. They are obtainable from the online motion database [4]. In order to evaluate the efficacy of our method, the experiments were performed based on VCA motion ranging from basic human movements (running and walking) to actions of higher complexity such as the cartwheeling and swordplaying procedures. It is worth noting that the proposed algorithm

Table 5.1: Results for the compression performance of the VCA mapping technique using k-mean clustering. Here, *FN* - denotes the motion index, *Desc* - denotes the motion description, *AMC* - original motion data file size in (KB), *Fr* - number of frames, *Ns* - denotes number of segment, *Cs* - the compressed file size of the motion file (KB) using J2K, *DEF* - Displacement Error per frame, *CR* - compression ratio of uncompressed to compressed motion files.

<i>FN</i>	<i>Desc</i>	<i>AMC(KB)</i>	<i>Fr</i>	<i>Ns</i>	<i>Cs(KB)</i>	<i>DEF</i>	<i>Cr</i>
02_01	<i>walk</i>	267.0352	343	6	34.6201	0.0063	7.7133
02_03	<i>run_jog</i>	135.0176	173	3	17.2705	0.0119	7.8178
02_04	<i>jump_bal</i>	380.3223	483	8	49.7725	0.0036	7.6413
02_05	<i>punch</i>	1449.2314	1854	30	186.5615	0.0028	7.7681
02_07	<i>swordplay</i>	1749.8291	2251	37	226.4277	0.0043	7.7280
02_10	<i>washself</i>	2056.2861	2645	43	275.6318	0.0020	7.4603
49_01	<i>walk</i>	510.3652	652	11	49.9268	0.0039	7.2986
49_05	<i>run_leap</i>	127.8887	164	3	17.4336	0.0128	7.3358
49_06	<i>cartwheel</i>	375.8672	481	8	50.5732	0.0088	7.4321
49_14	<i>dance</i>	484.0762	619	10	66.1406	0.0007	7.3189
49_19	<i>balance</i>	748.9629	957	16	104.3301	0.0005	7.1788
49_21	<i>acrobatic</i>	1888.5732	2422	40	261.7939	0.0038	7.2155

extends to other formats including the MPEG-4 BAP which can be used as the common representation layer for the motion capture information, thus the proposed method is not only constraint to the existing data representation tested.

5.3.1 VCAI using K-mean clustering

Table 5.1 presents the compression and error performances of the VCA motion data using VCA mapping technique using K-mean clustering. Here, *FN* denotes the original motion capture filename, *Desc* is the description of the motion sequence and *AMC* represents the AMC file size in (KBytes). The total frames count and compressed file size of clustered VCA data is denoted as *Fr* and *Cs*. The displacement error per frame (Eq. 5.7) for all sequences are shown in *DEF* of Table 5.1. Table 5.1 demonstrates that the VCAI (k-mean) mapping technique is able to achieve an average of 7.4923 compression ratio easily against the

Table 5.2: Results for the compression performance of the VCA mapping technique using FCM. Here, FN - denotes the motion index, Desc - denotes the motion description, AMC - original motion data file size in (KB), Fr - number of frames, Ns - denotes number of segment, Cs - the compressed file size of the motion file (KB) using J2K, DEF - Displacement Error per frame, D_{norm} - Normalized Distance, CR - compression ratio of uncompressed to compressed motion files.

<i>FN</i>	<i>Desc</i>	<i>AMC(KB)</i>	<i>Fr</i>	<i>Ns</i>	<i>Cs(KB)</i>	<i>DEF</i>	<i>D_{norm}</i>	<i>Cr</i>
02_01	<i>walk</i>	267.0352	343	6	34.8057	0.0054	0.0000798	7.6722
02_03	<i>run_jog</i>	135.0176	173	3	17.7705	0.0088	0.0001262	7.5979
02_04	<i>jump_bal</i>	380.3223	483	8	51.4658	0.0028	0.0000388	7.3898
02_05	<i>punch</i>	1449.2314	1854	30	187.3984	0.0027	0.0001170	7.7334
02_07	<i>swordplay</i>	1749.8291	2251	37	228.8145	0.0040	0.0000506	7.6474
02_10	<i>washself</i>	2056.2861	2645	43	275.0254	0.0021	0.0000380	7.4767
49_01	<i>walk</i>	510.3652	652	11	70.3086	0.0037	0.0001131	7.2589
49_05	<i>run_leap</i>	127.8887	164	3	16.8291	0.0137	0.0001302	7.5993
49_06	<i>cartwheel</i>	375.8672	481	8	49.8330	0.0089	0.0000793	7.5425
49_14	<i>dance</i>	484.0762	619	10	66.3301	0.0008	0.0000296	7.2980
49_19	<i>balance</i>	748.9629	957	16	105.9609	0.0004	0.0000307	7.0683
49_21	<i>acrobatic</i>	1888.5732	2422	40	263.2754	0.0036	0.0012000	7.1734
55_06	<i>chicken</i>	7248.5557	9275	150	975.1768	0.0009	0.0001980	7.4331
55_08	<i>monkey</i>	4155.1318	5340	87	535.6719	0.0030	0.0002918	7.7569
55_12	<i>dancingbear</i>	1618.6123	2075	34	203.2773	0.0026	0.0003162	7.9626
55_20	<i>panda</i>	3588.5010	4598	75	462.5059	0.0024	0.0001512	7.7588

original AMC file without any visual degradation in the reconstructed motion quality. The VCAI (k-mean) scheme performs the additional task of considering the DOFs coherence based on their allocated joints position, the similar DOFs motion data is iteratively clustered accordingly before compression.

In our simulation, it is shown that the proposed VCAI (k-mean) method maintained a constantly stable performance of 7.1 - 7.8 *Cr* for the different length of motion sequences of different complexity. This comparison shows that the VCAI (k-mean) is able to achieve competitive *Cr* and produces significantly improved error performance for near lossless compression against the earlier proposed VCAM and SVCA techniques.

Table 5.3: Results for the performance of un-clustered VCAI (Cs_1 , DEF_1 , $Dnorm_1$ and Cr_1 vs FCM technique(Cs_2 , DEF_2 , $Dnorm_2$ and Cr_2).

<i>Desc</i>	<i>Cs_1(KB)</i>	<i>Cs_2(KB)</i>	<i>DEF_1</i>	<i>DEF_2</i>	<i>Dnorm_1</i>	<i>Dnorm_2</i>	<i>Cr_1</i>	<i>Cr_2</i>
<i>walk</i>	35.4902	34.8057	0.0057	0.0054	0.0001974	0.0000798	7.5241	7.6721
<i>run_jog</i>	18.1631	17.7705	0.0095	0.0088	0.0003430	0.0001260	7.4336	7.5979
<i>jump_bal</i>	51.5859	51.4658	0.0029	0.0028	0.0000574	0.0000388	7.3726	7.3898
<i>swordplay</i>	230.3467	228.8145	0.0042	0.0040	0.0000840	0.0000506	7.5965	7.6474
<i>walk</i>	71.0371	70.3086	0.0038	0.0037	0.0001235	0.0001130	7.1845	7.2589
<i>run_leap</i>	17.0664	16.8291	0.0139	0.0137	0.0002862	0.0001302	7.4936	7.5993
<i>cartwheel</i>	50.3770	49.8330	0.0092	0.0089	0.0002822	0.0000793	7.4611	7.5425
<i>dance</i>	67.5723	66.3301	0.0008	0.0008	0.0000395	0.0000296	7.1638	7.2980
<i>chicken</i>	979.9561	975.1768	0.0010	0.0009	0.0002136	0.0001980	7.3968	7.4331
<i>dancingbear</i>	206.3369	203.277	0.0028	0.0026	0.0003921	0.0003160	7.8445	7.9626
<i>panda</i>	466.2148	462.5059	0.0026	0.0024	0.0005007	0.0001510	7.6971	7.7588

5.3.2 VCAI using Fuzzy-C mean clustering

Table 5.2 presents the compression and error performances of the VCA motion data using VCAI method for near lossless compression using fuzzy-c mean clustering. To solve (Eq. 5.9), the optimal result from 100 simulation cycles of each sequence is determined. Table 5.2 demonstrated that the VCAI (FCM) mapping technique is able to achieve an average of 7.5231 compression ratio against the original AMC file without any visible degradation in the reconstructed motion quality.

The proposed VCAI performs the additional task of considering the DOFs coherence based on their allocated joint positions. Thus, by using FCM, the similar DOFs motion data is iteratively clustered before compression. In [104], the compression ratio for different VCA motions are reported from 4.7 – 10 using the predictive based encoding method with average D_{norm} range of 0.21-0.221. The simulation showed that the proposed VCAI (FCM) method maintained a constantly stable performance of 7.0 - 8.0 Cr with average D_{norm} performances

Figure 5.2: Comparison results for the lossy compression of VCAI technique for sequence (a-b)02_01 and (c-d)02_03.

of 0.0001869 for the different length of motion sequences of different complexity which is significantly improved from the earlier quality performance using the k-mean algorithm. From the observation, the VCAI (FCM) encoding scheme produces high quality VCA sequences with little to non-perceivable perception quality deterioration of the skeletal model's motion upon lossless decoding.

Table 5.3 compared the DEF, D_{norm} and compression ratio of 11 different VCA motions against its unclustered counterpart. For consistency, the notation for parameters in the unclustered scheme is denoted to be Cr_1, DEF_1, Dnorm_1 and clustered VCAI method to be Cr_2, DEF_2 and Dnorm_2. The results in

Figure 5.3: Comparison results for the lossy compression of VCAI technique for sequence (a-b)49_05 and (c-d)49_06.

Table 5.3 show that the VCAI method outperforms its unclustered counterpart in term of both error and compression performance. On the average, the VCAI method is able to achieved an average of $Cr = 7.56$ with $DEF = 0.0049$ for different sequences as demonstrated in Table 5.2. The proposed VCAI (FCM) also did well against [36] where compression ratio of 1.7-2.5 for varying DEF of 0.17-14 is previously reported. To further reduce the data sizes needed to encode the VCA sequence, the compression of the *J2K* is varied from [2...20] to generate the lossy compressed stream for each motion sequence.

The rate distortion performances from the lossy compression techniques (FCM and un-clustered) are presented in Fig. 5.2 and Fig. 5.3. The simulation results

for 02_01(walking), 02_03(running), 49_05(run and leap) and 49_06(cartwheel) motion sequences are reported here. DEF_1(unclustered) and DEF_2(VCAI) denotes the displacement error per frame and Dnorm_1 and Dnorm_2 denote the normalized distance. Figure 5.2(a) and 5.2(c) show the simulation result of DEF vs Size(bytes) and data size (bytes) with its respective Cr . Figure 5.2(b) and 5.2(d) show the simulation result of Dnorm vs Size(bytes) and data size (bytes) with its respective Cr . In the figures, the primary axis denotes the error performance ((DEF), D_{norm}) and the secondary vertical axis denotes compression ratio (Cr). The experimental result against its unclustered counterpart is also provided in all figures. From the plots, it is observed that the (DEF, D_{norm}) value is reduced accordingly as the compression in the J2K encoder is increased. For a higher Cr , there is a proportional drop in error quality as determined by (DEF, D_{norm}).

The experimental result shows that the proposed VCAI method outperforms the unclustered scheme in term of both DEF and D_{norm} performances, consistently over different bit budget of the VCA. An interesting observation can be seen in Fig. 5.3(c,d) where the gain in DEF improvement is smaller in comparison to the others plots. This is due to the limited coherence presented in both structural and temporal data of the cartwheel sequence, thus decreasing the encoding efficiency of both algorithms. However in all cases, the VCAI (FCM) shows consistently better R-D performance for different VCA sequences with varying bit rate.

Lastly, Fig. 5.4 demonstrates the lossy performances of VCAI (MMF) for 02_01 and 02_03 sequences. The simulation results demonstrated that the pro-

Figure 5.4: Comparison results for the lossy compression of VCAI (MMF) against VCAI for sequence (a-b)02_01 and (c-d)02_03 are reported. The R-D plot from low to high Cr is presented in the figures. (a) and (c) denote the results from the original VCAI (FCM) method, (b) and (d) represent the improved experimental results with the VCAI (MMF).

posed VCAI technique coupled with the MMF is capable of achieving high compression ratios of 120:1 and 85:1 for both skeleton's motion sequences tested while maintaining a reasonable low $DEF \leq 15$. Without the MMF, this compression performance will result in $DEF \leq 20-25$ for the VCAI method. The reported results are also competitive against the observations reported in [104] where compression ratio (Cr) of up to 72:1 can be achieved and Cr of 1.7-37 in [26] for varying distortion.

The experiment shows that although it is technically possible to achieve a

higher $Cr \geq 200$ using our existing method, the subjective lost in quality by the decoded VCA is unacceptable visually during observation. The visual evaluation of the compressed stream demonstrated from the VCAI (MMF) showed perceptually minor foot skates and swaying on the VCA's motion where the resultant $DEF \leq 15$. Using the VCAI (MMF) method, this DEF is achievable at a Cr of approximately 120:1 for 02_01 walking sequence and 85:1 for 02_03 running sequence. In the case of $DEF \leq 10$ where only minor visual artifacts or near original motion sequence is obtained, our proposed method is able to achieve the desired quality at a competitive compression ratios of 100:1 and 60:1 on both tested sequences. Thus, making VCAI (MMF) suitable for low-powered mobile devices which need to satisfy the constraint of limited data capacity. The video presentation for the proposed VCAI (MMF) method across highly compressed range is available online at [15].

5.4 Conclusions and discussion

This Chapter reviews and presents a novel concept namely "Virtual Character Animation Image" (VCAI). In contrast to the previously discussed SVCA and VCAM method, the proposed scheme considers both the anatomical characteristics of the VC and inherent temporal coherence of the skeletal's motion, and jointly mapped the two characteristics into a VCAI prior compression. Coupled with a modified motion filter (MMF), which overcomes the limitation of 'Ring-ing' artifacts in image compression for VCA, the VCAI (MMF) provides highly compressed VCA at low quality degradation across varying bit rates. This is

demonstrated in the experimental results shown from Section 5.2.2, Table 5.2-5.4. The simulation results show that the proposed VCAI (MMF) is more efficient compared to the existing predictive and transform methods. In general, our method is able to achieve a competitive compression ratio of up to 120:1 for VCAI with minor visual artifacts. The VCAI outperforms the existing literature where compression ratio of up to 72:1 is reported for varying distortions.

One limitation faced during the research involves the need for prior analysis of the entire motion signal for normalization which could increase computing time. The solution to it involves partitioning the full VCA sequence into smaller motion segments. This allows the different motion blocks to be individually encoded and compressed to reduce the latency time and complexity incurred within the processing stages.

Figure 5.5: Reconstructed frames on varying motion sequences with the VCAI scheme. The individual sequence are clustered based on their DOFs similarities using FCM and compressed using the J2K standard. Both the motion temporal coherence and skeletal structure of the VC are exploited to improve compression efficiency and visual performance.

Chapter 6

Maximum Mutual Information for Virtual Character's Motion Compression

6.1 Introduction

The commonly use, pre-obtained motion capture sequences exhibit two distinct characteristics in the form of structural coherency and temporal correlations. In the previous chapter, we jointly considered the two unique VCA characteristics and exploit them for compression purposes. Existing, there is a need to further look into a common representation to provide proper analysis of the two correlations together.

6.1.1 Review on VCAI and Correlations

In the previous chapter, the VCAI technique demonstrated a new image based approach for the representation of VCA data and performs clustering to jointly exploit the correlations between the structural and temporal data of the VCA. In addition, the MMF (modified motion filter) was introduced to reduce the motion 'jittering' in VCA due to J2K [133] compression to further improve an-

imation quality. In comparison to previous literatures [37][104][26], the VCAI shows competitive improvement of compression performance demonstrated from experiments. However, with the direction of joint exploitation of both temporal and structural coherency possible, there is now a need to further investigate the two correlations using a common analytical tool. Mutual Information provides the means to determine the general dependencies between two sets of variable based on statistical sense. Previously, the applications for mutual information on speech recognition and transmission were studied in [81, 136]. Kim and Yook [81] presented the approach of combining linear spectral transformation with MMI (Maximum Mutual Information) to achieve rapid speech adaption. MI was used by Tsai and Lin [136] to devise the decision rules for transmission ordering. In [106], a joint saliency map-MI approach was proposed for the robust registration of images. Qin et al [106] uses the approach of joint saliency map to compare the saliency structures of two image to solve the problems of outliers and local maxima in MI-based image registration.

6.1.2 Contributions

Our proposed work differs from existing methods whereby we propose a new technique to analysis the joint correlations between the structural and temporal characteristics of the VCA. In this work, the use of MI is proposed to solve the dependency issue from a statistical approach. Both information and probability theories are considered to determine the correlation between the seemingly 'random' attributes in the VCA structure across frames. Mutual Information (MI) provides a measure to determine the statistical dependency between entropies in the DOFs across frames (Temporal) and among DOFs (Structural).

By maximizing the MI, the proposed VCA-MI rearranges the DOFs position on the image representation for best coherency prior to the image encoding process using JPEG2000. The proposed method uses image based encoding for the efficient representation of VCA data for reduction in processing complexity based on a $\mathbb{R}^2$ dimensionality. Thus, JPEG2000 and other advance image standards are applicable for our algorithm to further reduce the data size and improve flexibilities for transmission considerations.

6.1.3 Focus and Organization of Chapter

The focus of the Chapter is to discuss the implementation of a novel compression approach for Virtual Character Animation based on the criterion of maximization of mutual information in complex human motion sequences. The proposed scheme uses the concept of MI to provide a qualitative measure towards determining the inherent dependency between structural and temporal information presented in the skeletal human motions to achieve efficient compression.

The rest of the chapter is organized as follows. In Section 6.2, a review on both the Entropy and Mutual Information is presented. Section 6.3 provides the format and notation for the skeletal animation used for the rest of the Chapter. The algorithm for the determination of the VCA-MI is demonstrated in Section 6.4. Next, Section 6.5 describes the image based compression approach using the VCA-MI method. The simulation results of the proposed scheme are presented in Section 6.6. Finally, conclusions are presented in Section 6.7.

6.2 Entropy and Mutual Information

Information entropy describes the 'randomness' attributes in a collection of data. Shannon stated that with a higher entropy value between variables, the uncertainties over the variables also increases. Intuitively, the temporal characteristics of the VCA can be described using the Shannon's entropy concept.

$$\psi(Q) = - \sum_{\tau=1}^n p(q_{\tau}) \log_2 p(q_{\tau}) \quad (\text{Eq. 6.1})$$

$\psi(Q)$ denotes the entropy for q th DOF, where $\tau = 1, 2, \dots, n$. n is denoted as the number of possibilities for q_{τ} . $p(q_{\tau})$ describes the probability function. Mutual information measures the mutual dependency between two variables and has the advantage of maintaining invariancy across space transformation. The VCA sequences for human motion have two distinct characteristics in the form of structural coherence and temporal correlation. The MI's concept is applied to provide a systematic measure to describe the two seemingly 'random' sets of variables.

6.3 Parameters for skeletal based animation

The motion deviations within the joints of the skeletal avatar across frames are described using a collection of rotation and translation variables defined by μ_i^j . Here, i denotes the degree of freedom(DOF) within the skeletal model where $i = 1, 2, \dots, n$ and j indicates the frame numbers in the animation sequence for $j = 1, 2, \dots, m$. $\mu_i^j, \forall i, \forall j$ describes the entire animation. The rows of the matrix from $\mu_i^1, \mu_i^2, \dots, \mu_i^m$ contain the joint modifications of a single DOF across the frames and $\mu_1^j, \mu_2^j, \dots, \mu_n^j$ represent the DOFs within the single j th frame. The DOF matrix

is further partitioned into two separate sets, where $X = \mu_1^j, \mu_2^j, \dots, \mu_{n-\kappa}^j$ and $Y = \bar{X}$ denote the lossy compressed DOFs and preserved DOFs (lossless compressed) respectively, to ensure the unequal protection of more critical nodes (root and end nodes) in the structure of the VC. Let $X, Y \subseteq \xi$ and $X \cap Y = \phi$, where ξ is denoted as the required DOFs needed to fully decode the VCA sequence. In our work, the critical nodes which include the root nodes and end joints (left foot, right foot etc.) are separately encoded and preserved while the intermediary joint nodes undergoes higher lossy compression. The preservation of critical nodes reduces error propagation and occurrence of footskates and end joints errors commonly presented in skeletal based compression.

6.4 Determination of Maximum Mutual Information

(Eq. 6.1) determines the entropy for q th DOF. Let s be a second set of DOFs variable where its entropy is formulated as $\psi(S) = -\sum_{\tau=1}^{n'} p(s_\tau) \log_2 p(s_\tau)$. In the algorithm, the number of variables from both set q th and s th DOF are equal due to the segmentation process of image described in (Section 6.5). The number of outcomes n and n' will be dependent on the coherency within the sequence and quantization factor. The conditional entropy of Q after observing set S , where $((Q, S) \in X)$ is given by:

$$\begin{aligned}
 \psi(Q|S) &= -\sum_{s \in S} p(s) \left[\sum_{q \in Q} p(q|s) \log_2 p(q|s) \right] \\
 &= -\sum_{s \in S, q \in Q} p(s, q) \log_2 p(q|s) \\
 &= -\sum_{s \in S, q \in Q} p(s, q) \log_2 p(s, q) / p(s) \quad (\text{Eq. 6.2})
 \end{aligned}$$

$p(q|s)$ is the conditional probability of Q given S . Given both $\psi(Q)$ and $\psi(Q|S)$, the MI can also be formulated as:

$$MI(Q, S) = \psi(Q) - \psi(Q|S) \quad (\text{Eq. 6.3})$$

or similarly, it can be presented using the follow equation.

$$MI(Q, S) = \sum_{q \in Q, s \in S} P_{joint}(q, s) \log_2 \left(\frac{P_{joint}(q, s)}{P(q)P(s)} \right) \quad (\text{Eq. 6.4})$$

where $P_{joint}(q, s)$ is the joint probability distribution function, and $P(q)$ and $P(s)$ are the marginal probability functions for Q and S . From (Eq. 6.3), we measured MI as a displacement between the two distributions(Q, S). It follows that the mutual information is both non-negative and symmetry in nature where $MI(Q, S) \geq 0$ and $MI(Q, S) = MI(S, Q)$. Since $(Q, S) \in X$, and $|X| = n - \kappa$, it is necessary to maximize and re-sort each remaining DOF s according to their similarities prior compression. The optimization process for the VCA-MI is described in Alg. 1.

Input: VC motion data
Output: Determine DOF index (NI) using VCA-MI

```

begin
  Initialise :  $\{H, MI, Final\_MI, NI\} \leftarrow \emptyset; i, \alpha \leftarrow 0; j \leftarrow 1;$ 
  /* Calculate Entropy(H) for each DOF set */
  for  $\alpha \leftarrow 1$  to  $n - \kappa$  do
     $H(\alpha) = Cal\_Entropy(DOF_\alpha)$ 
  end
  /* Calculate MI for each DOF */
  for  $\alpha \leftarrow 1$  to  $n - \kappa$  do
    for  $\beta \leftarrow 1$  to  $n - \kappa$  do
      if  $(\alpha \neq \beta)$  then
         $MI \leftarrow Cal\_MI(H(\alpha), H(\beta))$ 
      end
    end
  end
  end
  /* Determine first DOF reference */
   $Final\_MI(j) \leftarrow func\_largest(MI), \forall \alpha;$  /* Store motion data
  for MI_max */
   $NI \leftarrow DOF\_alpha;$  /* Store Original DOF index for MI_max */
  /* Find remaining  $n - 1 - \kappa$  sets using Greedy Selection */
  for  $j \leftarrow 2$  to  $n - \kappa$  do
    for  $i \leftarrow 1$  to  $n - \kappa$  do
      if  $(i \notin NI)$  then
         $MI \leftarrow Cal\_MI(H(DOF_\alpha), H(i))$ 
      end
    end
    /* Determine next DOF reference */
     $Final\_MI(j) \leftarrow func\_largest(MI), \forall \alpha;$  /* Store next motion
    data with MI closest to MI_max */
     $NI \leftarrow DOF\_alpha;$  /* Store next Original DOF index for
    MI_max */
  end
end

```

Algorithm 1: Algorithm for determining the Maximum Mutual Information for VCA. The algorithm outputs the sorted DOFs index(NI) based on the correlations between structural and temporal measures of the VCA.

From each DOF class $\mu_i \in X$, determine the DOF with the best MI relationship with its complement classes. DOF_α is taken as the initial DOF prior the Greedy

based selection. The remaining $n - 1 - \kappa$ DOF indices can now be determined progressively using the fast algorithm. The resultant NI from the optimization process contains the new DOF indices re-sort according to their MI relationship with other DOFs. The final motion map is stored in $Final_MI$. Each DOF indices is encoded as a 6-bits integer with its preserved counterpart encoded using 4-bits each. The DOFs indices are transmitted prior the compression process to ease decoding complexity.

6.5 Image Compression for VCA-MI

The calculated motion map contains $n - \kappa$ DOF classes of joints modification variables. The new motion map is denoted as $\tilde{X}$ and compressed by JPEG2000 (J2K) encoder. In [69], it is demonstrated that the J2K scheme is suitable for both natural and geometrical images (GI) outperforming the JPEG [108] in peak-signal-noise-ratio (PSNR) over varying bit-rate. The same compression scheme is adopted in the presented framework. The compressed image map is denoted as M .

Putting the individual function together, the VCA-MI process converts a set of VCA joint modification information to a compressed image map where $f : X \rightarrow M \in \mathbb{R}^2$.

$$f = j \circ m \circ w \tag{Eq. 6.5}$$

where $\circ$ denotes function composition between w, m and j to convert the VCA joint modification information into a compressed image map. As a summary, function w converts the original VCA data to a set of matrices denoted by X

Figure 6.1: Framework for image segmentation of VCA data. The original VCA data is partitioned for both lossy and lossless compression. Segmentation for lossy components is performed to improve correlation of VCA sequence.

for processing. $m : X \rightarrow \tilde{X}$ represents the mutual information maximization step to obtain the image map $\tilde{X}$. Lastly, the image map is compressed using the J2K standard as shown in $j : \tilde{X} \rightarrow M$. The compressed VCA image M is more flexible and robust than its original matrix representation X .

One important consideration during the VCA-MI encoding lies in the length of the motion sequence. Since the correlation of the VCA sequence have an inverse relation with the length of the sequence. The initial matrix $X = \{\mu_i^j, \forall i, \forall j\}$ is partitioned into l image segments of μ_τ^τ where $\tau = n - \kappa$. Similar to using macroblocks in image/video coding, making use of multiple image maps for the skeletal motion sequences enable us to obtain better quality VCA sequence with lessen complexity throughout (Eq. 6.5). Figure 6.1 demonstrates the framework for the image partitioning for both lossless and lossy components of the original VCA data. Both components are further segmented to improve correlation across the VCA sequence. To measure the quality of the compressed VCA sequence M , there is a need to define an error metric that can quantify the nearness between

the original X and compressed M of for VCA sequence. Since the accuracy of the DOF joint modifications does not necessary translates to the closeness between the joint vertices of the two. A better selection for the measurement should be the original coordinates of the joints against the reconstructed joints. The original joints coordinates are denoted as $J_{ij} = \{C_{ij}^x, C_{ij}^y, C_{ij}^z, \forall i, \forall j\}$ where C is the individual set of coordinate in its respective axis, i and j describe the frame number and DOF here. In our experiment, the root-mean-square error d_r is used as the metric to determine the quality of the decoded VCA against the original. d_r is defined as follows:

$$d_r(J, J') = \sqrt{\frac{1}{N} \sum_{i=1}^N \|J(C_{ij}) - J'(C'_{ij})\|^2} \quad (\text{Eq. 6.6})$$

J is the original joint coordinates and J' is denoted as the decoded coordinates. N represented the total number of joint nodes. The root mean square provided an objective measure to determine the quality degradation of the decoded VCA against the its original counterpart.

6.6 Experimental Results

Simulations were carried out to evaluate the VCA-MI performances for 11 different human motion sequences with varying actions complexity. Each test sequence contains 62 degree of freedoms (DOFs) and 31 joint nodes. The original motion test sequences were obtained from the online database [4]. Although the Acclaim skeleton/motion capture file(ASF/AMC) was used during the experiment, our method is also extendable to other VCA formats making it versatile for generic human motion sequences.

Table 6.1: Root mean square(RMS) errors of different test motion capture data using the VCA-MI approach. Here, Orig_Size(KB) - original motion capture AMC file size in (KBytes), Segs - number of segments, Comp_Size - Compressed file size of VCA data, dr - Root mean square Error, Cr - Compression ratio

<i>Motion Description</i>	<i>Orig_Size(KB)</i>	<i>Segs</i>	<i>Comp_Size(KB)</i>	<i>dr</i>	<i>Cr</i>
<i>walking</i>	267.0352	6	37.9746	0.000183	7.0319
<i>hoop scotch</i>	292.8857	6	41.2705	0.000139	7.0967
<i>Long Jump</i>	390.3164	8	53.6338	0.000282	7.2774
<i>Kick ball(Action_1)</i>	458.6221	9	65.6396	0.000175	6.9870
<i>Kick ball(Action_2)</i>	281.3896	5	40.4512	0.000201	6.9563
<i>Jump/Kick/Spin</i>	480.7861	9	66.7588	0.000392	7.2018
<i>Backflip</i>	193.4609	3	26.9355	0.001100	7.1824
<i>Cartwheeling</i>	396.0635	8	56.6201	0.000690	6.9951
<i>Dance(Action_1)</i>	709.9121	14	101.831	0.000274	6.9715
<i>Dance(Action_2)</i>	457.5322	9	64.7559	0.000467	7.0655
<i>Dance(Action_3)</i>	408.9502	8	57.9756	0.000530	7.0538

Table 6.1 presents the near lossless compression results using the VCA-MI approach. Org_Size represents the original amc data size and comp_size is the compressed size in (KB). dr denotes the root mean square error obtained from Eq. 6.6. During the experiment, each test sequence is partitioned into two independent matrix $\mu_1^j, \mu_2^j, \dots, \mu_{n-\kappa}^j$ (inter joint modifications) and $\mu_1^j, \mu_2^j, \dots, \mu_\kappa^j$ (crucial nodes) for compression. In order to ensure the preservation of crucial end joints coordinates, $\mu_1^j, \mu_2^j, \dots, \mu_\kappa^j$ were lossless compressed to prevent substantial loss in quality to the VCA. The inter joints matrices $\mu_1^j, \mu_2^j, \dots, \mu_{n-\kappa}^j$ are optimized using the MI approach explained in Sect. 6.4 before both matrices are compressed and re-combined at the decoder to obtain the final VCA sequence. The results show that VCA-MI achieve an average rms of 0.00304 and Cr of 7.0798 for near lossless setting across 11 test human motion sequences of varying complexity. Table 6.2 shows the simulation results for VCA-MI across different compression ratio. Q is denoted as the rate values or factor of compression. For the experiment, the

Table 6.2: Results for the VCA-MI performances across varying compression rates. Here, *RMS* denotes the Root mean square error. *Q* is the varying compression rate value of J2K encoder. *Cs* and *Cr* are the Compressed file size(KBytes) and Compression ratio respectively.

VCA-MI performances over varying Compression ratio -J2K (<i>Q</i>)							
Motion Description	Readings	<i>Q</i> =1	<i>Q</i> =2	<i>Q</i> =3	<i>Q</i> =4	<i>Q</i> =5	<i>Q</i> =6
Hop Scotch	RMS	0.0001	0.0035	0.0201	0.0417	0.0718	0.1006
	Cs	41.2705	27.6846	21.9482	19.1768	17.5088	16.4443
	Cr	7.0967	10.5794	13.3444	15.2730	16.7279	17.8107
Long Jump	RMS	0.0003	0.0168	0.0300	0.0719	0.1125	0.1560
	Cs	53.6338	36.7031	29.5908	25.7949	23.6074	22.1387
	Cr	7.2774	10.6344	13.1905	15.1315	16.5336	17.6305
KickBall(Action_1)	RMS	0.0002	0.0053	0.0283	0.0622	0.1007	0.1349
	Cs	65.6396	43.3584	34.3867	29.9229	27.2656	25.4199
	Cr	6.9870	10.5775	13.3372	15.3268	16.8205	18.0418
KickBall(Action_2)	RMS	0.0002	0.00630	0.0380	0.0854	0.1439	0.2032
	Cs	40.4512	26.7998	21.1748	18.4346	16.7666	15.6807
	Cr	6.9563	10.4997	13.2889	15.2642	16.7827	17.9450
Jump/Kick/Spin	RMS	0.0004	0.0065	0.0367	0.0769	0.1268	0.1788
	Cs	66.7588	44.8496	35.5000	30.5215	27.9063	26.0479
	Cr	7.2018	10.7200	13.5433	15.7524	17.2286	18.4578
Backflip	RMS	0.0011	0.0113	0.0594	0.1434	0.2308	0.3147
	Cs	26.9355	18.1406	14.3867	12.4629	11.3584	10.5713
	Cr	7.1824	10.6645	13.4472	15.5230	17.0324	18.3006
Cartwheel	RMS	0.0007	0.0116	0.0683	0.1455	0.2410	0.3507
	Cs	56.6201	37.6709	29.7148	25.9492	23.5947	22.0527
	Cr	6.9951	10.5138	13.3288	15.2630	16.7861	17.9598
Dance(Action_1)	RMS	0.0003	0.0092	0.0487	0.1204	0.1875	0.2761
	Cs	101.8311	67.1797	53.3232	46.1533	42.1064	39.3525
	Cr	6.9715	10.5674	13.3134	15.3816	16.8599	18.0398
Dance(Action_2)	RMS	0.0005	0.0082	0.0459	0.1020	0.1648	0.2326
	Cs	64.7559	43.2051	34.2539	29.8232	27.1045	25.2842
	Cr	7.0655	10.5898	13.3571	15.3415	16.8803	18.0956

Table 6.3: Comparison of VCA-MI against VCAI for lossless compression. Here, *DEF* denotes the displacement error per frame and *RMS* denotes the Root mean square error.

<i>Motion</i>	<i>DEF</i> (MI)	<i>DEF</i> (VCAI)	<i>RMS</i> (MI)	<i>RMS</i> (VCAI)	<i>DEF</i> %	<i>RMS</i> %	<i>Cs</i> %
<i>walking</i>	0.0044	0.0054	0.000183	0.000217	18.51	15.76	9.10
<i>run/jog</i>	0.0067	0.0088	0.000272	0.000359	23.86	24.20	8.82
<i>swordplay</i>	0.0027	0.0040	0.000152	0.000231	32.50	34.26	7.91
<i>run/leap</i>	0.0104	0.0137	0.000409	0.000556	24.08	26.45	9.02
<i>Cartwheel</i>	0.0051	0.0089	0.000254	0.000486	42.69	47.75	7.24

error performances for 9 motion sequences across increasing Q (J2K compression ratio) is recorded in Table 6.2. The result shows that the VCA-MI method is capable of ensuring the smooth degradation of the VCA quality determined by d_r as Q is progressively increased. The proposed scheme outperforms VCAI method demonstrating an improved d_r of 0.000139-0.0683 for Cr of 6.9563-7.2774 for nearly lossless compression. Subjectively, the decoded VCA is observed to be near identical to the original motion sequence. For higher compression ratio of $Q = 5$ and $Q = 6$, slight jittering in the motion sequence can be observed. However, the jittering translated as noise from the J2K encoder can be easily removed using the MMF filter proposed from [44]. From our observation, the VCA-MI is more suitable for VCA compression for near lossless compression due to the need for added preservation of the end joints.

Lastly, the experimental comparison against VCAI for 5 sequences of varying complexity is presented in Table 6.3. $DEF\%$ and $RMS\%$ denotes the percentage improvement for DEF and RMS in comparison to VCAI. $Cs\%$ shows the percentage increase in data size in comparison to VCAI. The experimental result shows an improvement in both the RMS and DEF performance of the proposed VCA-MI against VCAI for lossless compression. This is reported at an average improvement of 28.3% and 29.6% for both DEF and RMS across the 5 sequences during lossless compression. From our observation, the improvement in VCA quality comes with a slight tradeoff in the increase of data size, at an average of 8.4% over the 5 sequences. For VCA-MI, quality preservation is the focus of the algorithm. Thus unlike [44], the DOFs are first separated based on their

criticality and encoded differently using more bits for the more important nodes (roots and end joints) for better VCA reconstruction.

6.7 Conclusions

In this Chapter, a systematic approach towards the compression of virtual human skeletal animation is presented. Using the criterion of maximization of mutual information, a qualitative measurement for determining the inherent dependency between the skeletal model hierarchical behavior and motion temporal characteristic of the VCA is shown. Experimental results demonstrated the compression and error performances for the proposed scheme over different VCA sequences for varying levels of complexity.

Chapter 7

Conclusions and Future Work

In this Chapter, we conclude the studies presented in this thesis and propose some suggestions for the future work.

7.1 Conclusions

Chapter 3 presented the studies of an image based encoding method for the representation of static 3D model. The discussed method is applicable for the conventional broadcasting standard. The problem of delivering progressively encoded 3D content across packet erasure channels was investigated. The chapter introduced a novel 3D encoding method, Spectral Geometry Image (SGI), which is more robust, and efficient in comparison to the state of art Geometry Image technique for 3D models representation. It showed that by coupling SGI together with the proposed joint source and channel coding (JSCC) scheme, an effective framework for delivery of progressively encoded 3D model can be realized. Experimental results demonstrated this fact and showed that the proposed method outperforms the conventional GI in terms of coding efficiency and error resilience performance simulated for varying packet lost rates.

Chapter 4 was dedicated to the studies of two proposed encoding schemes for the widely used skeletal avatar motion data. Two aspects of the VC encoding namely (1) Progressive encoding of VCA and (2) Efficient compression of the VCA were studied.

The Scalable virtual character animation (SVCA) was presented to achieve efficient scalable transmission of virtual character animation using the unique parent/child relationship presented within structural hierarchy of the VCA. The proposed algorithm addressed the issue of flexibility in bit-stream for delivering virtual character animation to low-power mobile devices and bandwidth constraint applications. In the second scheme, Virtual Character Animation Mapping (VCAM) demonstrated the encoding of VCA information using images to achieve efficient compression. The temporal characteristic of the motion information is utilized as spatial redundancies of an image and further compressed using the JPEG2000 standard. Experimental results showed that the proposed method have competitive performance compared to the SVCA algorithm for rate-distortion performance.

In Chapter 5, the challenges and solutions to jointly utilized the two distinct characteristics of the VCA were studied. The Virtual Character Animation Image (VCAI) addressed the mentioned problem using an image based encoding approach for the representation of VCA data. Built upon a fuzzy clustering algorithm, the data similarity within the anatomy structure of a Virtual Character (VC) model was jointly considered with the temporal coherence within the motion data to achieve efficient data compression. Since the VCA is mapped as an image, the use of image processing tool is possible for efficient compression

and delivery of such content across dynamic network. A modified motion filter (MMF) was also proposed to minimize the visual discontinuity in VCA's motion due to the quantization and transmission errors. The MMF helps to remove high frequency noise components and smoothen the motion signal providing perceptually improved VCA with lessened distortion. Simulation results showed that the proposed algorithm is competitive in compression efficiency and decoded VCA quality against the state-of-the-art VCA compression methods, making it suitable for providing quality VCA animation to low-powered mobile devices.

In Chapter 6, the need for a systematic approach towards the compression of virtual human skeletal animation was discussed. The use of the criterion of maximization of mutual information provided a qualitative measurement for determining the inherent dependency between the skeletal model hierarchical behavior and motion temporal characteristic of the VCA. MI approaches the difficult problem of finding dependencies between the motion and structure of a VCA from a statistical measure between the two correlations. Coupled with the state of art VCAI method shown in Chapter 5 to encode the VCA animation into images for compression, good quality VCA sequence with efficient compression can now be achieved for varying motion sequence with different complexity.

7.2 Future Work

Based on the studies of the thesis, the following recommendations are made as possible research directions in the future.

7.2.1 Extension of SGI research

First, the transmission of SGI across wireless channels poses a challenging problem due to the erratic and lossy nature of wireless channels. There is a need to develop better error resilient, correction and concealment tools to ensure the smooth delivery of 3D model across the error prone network. Next, conformal parameterization to a rectangular domain may introduce very large area distortion if the 3D model has complicated topology and geometry. In the future, we are going to generalize the spectral geometry image to the polycube domain [70, 140] and apply it to real-world models of complicated geometry and topology.

7.2.2 Progressive transmission of 3D animation using SGI

The demand for 3D animated data is growing rapidly in recent years due to the technological improvement in graphical data handling capabilities and the mass usage of such representations in the area of digital entertainment and engineering simulations. Several early researches in 3D were directed to the studies of static based models including the areas of compression and progressive transmission of such data. However, the direct application of existing state-of-art static based 3D techniques on animated meshes might not be as efficient since the temporal coherency between frames is not considered. This may result in topological and motion inconsistency causing disjoint and poor decoded animated movement across frames. For future works, we intend to carry out studies to exploit the image characteristics of the SGI to address the problems in assuring motion coherence and topological consistency for such data representation.

7.2.3 Error resilient transmission of 3D animated meshes

The transmission of progressive 3D animation across noisy channel is a relatively new area of research and the need is significant with the rapid growth in demand to share 3D animated contents across network. Currently, the focus for 3D animation research is centered on the data reduction. There is a need to address the efficient transmission for such content across channels with heterogeneous characteristics and providing flexibility for the bit-stream to be partially decodable at required bit rates to achieve the optimized reconstructed animation quality. Another research direction involves the implementation of dedicated error control schemes for the resilient transmission of such contents. Error control techniques such as Transport Level Error Control and Error Resilience can be further investigated.

7.2.4 Segmentation of VCA for efficient motion analysis

Currently, the VCA-MMF scheme introduced in Chapter 5 demonstrates good compression performances for VCA sequences with distinction motion variations. In the event that a single VCA sequence is made up of several complex motions performed in quick succession across time, the varying motion patterns could cause correlations between frames to become inconsistent and thus affects the VCA-MMF's performances. Motion segmentation for VCA is a relatively new area of research whereby it enables clear identification and distinct segregation of behaviors and motion patterns within VCA sequences. The studies for having the motion segmentation scheme as a pre-process for the VCA-MMF should be

further explored to ensure consistent compression and quality performances for such VCA sequences.

7.2.5 Efficient representation and compression of VCA-MMI data

MMI approach was introduced in Chapter 6 to determine the scope of dependency between the structure of VCA model and temporal motion in the animation. The current limitation lies in the need to reserve additional data bits for the encoding of key DOF parameters. This limits the compression performance of the VCA whereby the reserved DOF compressed using lossless technique will form the bottleneck for the scenario of highly compressed VCA. There is a need to explore more efficient techniques for the representation/compression of such data to address the limitation and customizing the algorithm for both lossless and lossy compression.

Author's Publications

Transactions and Conferences

- (1) B. S. Chew, L. P. Chau, Y. He, D. Wang, Steven C.H. Hoi, Spectral Geometry Image: Image Based 3D Models for Digital Broadcasting Applications, IEEE Transactions on Broadcasting, vol.57, no.3, pp.636-645, Sept. 2011.
- (2) B. S. Chew, L. P. Chau and K. H. Yap, A Fuzzy clustering algorithm for Virtual Character Animation representation, IEEE Transactions on Multimedia, Volume 13, No. 1, pp. 40-49, February 2011.
- (3) B. S. Chew, L. P. Chau and K. H. Yap, Maximum Mutual Information for Virtual Characters Motion Compression, to be submitted to IET Image Processing, 2012
- (4) B. S. Chew, L.P. Chau and K. H. Yap, Image Based Approach with K-Mean Clustering for the Compression of Human Motion Sequences, IEEE International Symposium on Circuits and Systems, ISCAS 2011, pp. 1964-1967, Rio de Janeiro, Brazil, 16-18 May 2011
- (5) B. S. Chew, L.P. Chau and K. H. Yap, Virtual Character Animation Mapping, International Conference on Information, Communications & Signal Processing, ICICS 2009, pp. 1-4, Macau, December 2009
- (6) Y. He, B. S. Chew, D. Wang, Steven C.H. Hoi and L. P. Chau, Streaming 3D Meshes Using Spectral Geometry Images, ACM international conference on Multimedia, pp. 431-430, Beijing, China, October 19-23, 2009

- (7) B. S. Chew, L.P. Chau and K. H. Yap, Progressive Transmission of Motion Capture Data for Scalable Virtual Character Animation, IEEE International Symposium on Circuits and Systems, ISCAS2009, pp.1461-1464, Taipei, Taiwan, May 2009
- (8) B. S. Chew, L.P. Chau and K. H. Yap, An efficient Error Protection Scheme for Point Based 3-D models over Packet Erasure Network, 2009 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB2008, pp. 1-4, Bilbao, Spain, May 2009
- (9) B. S. Chew, L.P. Chau and K. H. Yap, Bitplane Coding Technique for 3-D Animated Meshes, 2008 IEEE International Conference on Neural Networks and Signal Processing (ICNNSP08), pp. 692-695, Zhenjiang, China, June 2008

References

- [1] Coding of Audio-Visual Objects: Scene description and application engine (BIFS, XMT, MPEG-J). ISO/IEC 14496-11.
- [2] ACCLAIM ASF/AMC FORMAT DESCRIPTION. [online]. Available: <http://research.cs.wisc.edu/graphics/courses/cs-838-1999/jeff/asf-amc.html>.
- [3] Avatar: Official website for movie Avatar. [online]. Available: <http://www.avatarmovie.com/>.
- [4] CMU Graphics Lab Motion Capture Database. [online]. Available: <http://mocap.cs.cmu.edu/>.
- [5] Coding of Audio-Visual Objects: Animation Framework EXtension (AFX), ISO/IEC standard 14496-16, 2002.
- [6] Coding of Audio-Visual Objects: Systems, ISO/IEC standard 14496-1, 1998.
- [7] Fine granularity scalability with wavelets coding, ISO/IEC JTC1/SC29/WG11,MPEG98/M4021, Oct.1998.
- [8] H-Anim 1.1. Humanoid Animation Working Group. [online]. Available: <http://h-anim.org/specifications/h-anim1.1/>.
- [9] Matching pursuits residual coding for video fine granular scalability, ISO/IEC JTC1/SC29/WG11,MPEG98/M3991, Oct.1998.

- [10] MPEG-21: The multimedia framework. [online]. Available: <http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm>.
- [11] MPEG: The standard for multimedia for the fixed and mobile web. [online]. Available: <http://mpeg.chiariglione.org/standards/mpeg-4/mpeg-4.htm>.
- [12] MPEG: The Moving Picture Experts Group. [online]. Available: <http://mpeg.chiariglione.org/>.
- [13] MPEG-4 Animation Framework eXtension Overview. [online]. Available: <http://mpeg.chiariglione.org/technologies/mpeg-4/mp04-afx/index.htm>.
- [14] Openjpeg library. An open source JPEG2000 codec. [online]. Available: <http://www.openjpeg.org/>.
- [15] VCAI. [online]. Available: <http://web.mysites.ntu.edu.sg/n060035/publicsite/shared%20documents/forms/allitems.aspx>.
- [16] Coding of Audio-Visual Objects: Visual. ISO/IEC 14496-2. July 2001.
- [17] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete Cosine Transform. *IEEE Transactions on Computers*, C-23(1):90–93, Jan. 1974.
- [18] G. Al-Regib, Y. Altunbasak, and J. Rossignac. A joint source and channel coding approach for progressively compressed 3-D mesh transmission. In *IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 161–164, 2002.
- [19] Ghassan Al-Regib and Yucel Altunbasak. 3tp: 3-d models transport protocol. In *Proceedings of the ninth international conference on 3D Web technology, Web3D '04*, pages 155–162, New York, NY, USA, 2004. ACM.
- [20] Ghassan Al-Regib, Yucel Altunbasak, and Jarek Rossignac. An unequal error protection method for progressively compressed 3-D meshes. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 2041–2044, May 2002.

- [21] Ghassan Al-Regib, Yucel Altunbasak, and Jarek Rossignac. Error-resilient transmission of 3d models. *ACM Trans. Graph.*, 24(2):182–208, 2005.
- [22] Marc Alexa and Wolfgang Mller. Representing animations by principal components. *Computer Graphics Forum*, 19(3):411–418, 2000.
- [23] Pierre Alliez and Mathieu Desbrun. Progressive compression for lossless transmission of triangle meshes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 195–202, 2001.
- [24] Pierre Alliez and Mathieu Desbrun. Valence-driven connectivity encoding for 3d meshes. *Computer Graphics Forum*, 20(3):480–489, 2001.
- [25] Pierre Alliez and Craig Gotsman. Recent advances in compression of 3d meshes. In *Advances in Multiresolution for Geometric Modelling*, pages 3–26. Springer-Verlag, 2003.
- [26] Okan Arikan. Compression of motion capture databases. *ACM Trans. Graph.*, 25:890–897, July 2006.
- [27] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. Mesh: measuring errors between surfaces using the hausdorff distance. In *IEEE International Conference on Multimedia and Expo, 2002. ICME '02. Proceedings.*, volume 1, pages 705 – 708, 2002.
- [28] M. Aviles and F. Moran. Static 3D triangle mesh compression overview. In *15th IEEE International Conference on Image Processing, ICIP 2008.*, pages 2684 –2687, Oct. 2008.
- [29] Alan H. Barr, Bena Currin, Steven Gabriel, and John F. Hughes. Smooth interpolation of orientations with angular velocity constraints using quaternions. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '92, pages 313–320, 1992.

- [30] M.O. Bici and G.B. Akar. Multiple description scalar quantization based 3d mesh coding. In *Proc. IEEE Int. Conf. Image Proc., ICIP 2007*, pages 553–556, Oct. 2006.
- [31] M.O. Bici, A. Norkin, and G.B. Akar. Packet loss resilient transmission of 3d models. In *Proc. IEEE Int. Conf. Image Processing, ICIP 2007*, pages 121 –124, Oct 2007.
- [32] M.O. Bici, A. Norkin, G.B. Akar, A. Gotchev, and J. Astola. Multiple description coding of 3d geometry with forward error correction codes. In *Proc. of 3DTV Conf.*, pages 1–4, May 2007.
- [33] Jules Bloomenthal. Modeling the mighty maple. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques, SIGGRAPH '85*, pages 305–311, 1985.
- [34] Samuel R. Buss and Jay P. Fillmore. Spherical averages and applications to spherical splines and interpolation. *ACM Trans. Graph.*, 20:95–126, April 2001.
- [35] Lei Cao. On the unequal error protection for progressive image transmission. *IEEE Transactions on Image Processing*, 16(9):2384 –2388, Sep. 2007.
- [36] S. Chattopadhyay, S. M. Bhandarkar, and K. Li. Model-based power aware compression algorithms for MPEG-4 virtual human animation in mobile environments. *IEEE Transactions on Multimedia*, 9(1):1 –8, Jan. 2007.
- [37] S. Chattopadhyay, S.M. Bhandarkar, and K. Li. BAP sparsing: A novel approach to MPEG-4 body animation parameter compression. In *Systems Communications, 2005. Proceedings*, pages 104 – 109, Aug. 2005.
- [38] S. Chattopadhyay, S.M. Bhandarkar, and Kang Li. Human motion capture data compression by model-based indexing: A power aware approach. *IEEE*

- Transactions on Visualization and Computer Graphics*, 13(1):5–14, Jan.-Feb. 2007.
- [39] Wei Cheng. Streaming of 3d progressive meshes. In *Proc. ACM Multimedia*, pages 1047–1050, NY, USA, 2008.
 - [40] Wei Cheng, Wei Tsang Ooi, Sebastien Mondet, Romulus Grigoras, and Géraldine Morin. An analytical model for progressive mesh streaming. In *Proc. ACM Multimedia*, pages 737–746, NY, USA, 2007.
 - [41] Wei Cheng, Wei Tsang Ooi, Sebastien Mondet, Romulus Grigoras, and Géraldine Morin. Modeling progressive mesh streaming: Does data dependency matter? *ACM Trans. Multimedia Comput. Commun. Appl.*, 7:10:1–10:24, March 2011.
 - [42] Won-Sik Cheong, Jihun Cha, Sangwoo Ahn, Won-Hyuck Yoo, and Kyung Ae Moon. Interactive terrestrial digital multimedia broadcasting(T-DMB) player. *IEEE Transactions on Consumer Electronics*, 53(1):65–71, Feb. 2007.
 - [43] Boon-Seng Chew, Lap-Pui Chau, and Kim-Hui Yap. Progressive transmission of motion capture data for scalable virtual character animation. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 1461–1464, May 2009.
 - [44] Boon-Seng Chew, Lap-Pui Chau, and Kim-Hui Yap. A fuzzy clustering algorithm for virtual character animation representation. *IEEE Transactions on Multimedia*, 13(1):40–49, Feb. 2011.
 - [45] Bennett Chow and Feng Luo. Combinatorial ricci flows on surfaces. *J. Differential Geom.*, 63(1):97–129, 2003.
 - [46] M.M. Chow. Optimized geometry compression for real-time rendering. In *Proceedings of Visualization '97*, pages 347–354, Oct. 1997.

- [47] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. Metro: measuring error on simplified surfaces. Technical report, Paris, France, 1996.
- [48] Daniel Cohen-Or, David Levin, and Ofir Remez. Progressive compression of arbitrary triangular meshes. In *Proceedings of the conference on Visualization '99: celebrating ten years, VIS '99*, pages 67–72, 1999.
- [49] Michael Deering. Geometry compression. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, SIGGRAPH '95*, pages 13–20, New York, NY, USA, 1995. ACM.
- [50] Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomír Měch, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques, SIGGRAPH '98*, pages 275–286, 1998.
- [51] O. Devillers and P.-M. Gandoin. Geometric compression for interactive transmission. In *Proceedings of Visualization 2000.*, pages 319 –326, Oct. 2000.
- [52] F. Dufaux, G.J. Sullivan, and T. Ebrahimi. The JPEG XR image coding standard [Standards in a Nutshell]. *IEEE Signal Processing Magazine*, 26(6):195–199, 204 –204, Nov. 2009.
- [53] Nira Dyn, David Levine, and John A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graph.*, 9:160–169, April 1990.
- [54] E. O. Elliott. A model of the switched telephone network for data communications. *Bell syst. Techn. J.*, pages 89–109, Jan. 1965.
- [55] Pierre-Marie Gandoin and Olivier Devillers. Progressive lossless compression of arbitrary simplicial complexes. In *Proceedings of the 29th annual*

- conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 372–379, 2002.
- [56] Pierre-Marie Gandoïn and Olivier Devillers. Progressive lossless compression of arbitrary simplicial complexes. *ACM Trans. Graph.*, 21:372–379, July 2002.
 - [57] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 209–216, 1997.
 - [58] Thomas Di Giacomo, Chris Joslin, Stphane Garchery, HyungSeok Kim, and Nadia Magnenat-Thalmann. Adaptation of virtual human animation and representation for MPEG. *Computers and Graphics*, 28(4):485 – 495, 2004.
 - [59] F. Sebastin Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3:29–48, March 1998.
 - [60] Xianfeng Gu, Steven J. Gortler, and Hugues Hoppe. Geometry images. In *SIGGRAPH*, pages 355–361, 2002.
 - [61] Xianfeng Gu, Sen Wang, Junho Kim, Yun Zeng, Yang Wang, Hong Qin, and Dimitris Samaras. Ricci flow for 3d shape analysis. In *ICCV*, pages 1–8, 2007.
 - [62] Xianfeng Gu and Shing-Tung Yau. Global conformal parameterization. In *Symposium on Geometry Processing*, pages 127–137, 2003.
 - [63] Y. Gu and W.T. Ooi. Packetization of 3d progressive meshes for streaming over lossy networks. In *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*, pages 415 – 420, oct. 2005.

- [64] Sir W. R. Hamilton. *Lectures on Quaternions*. Hodges Smith and Co., Dublin, 1853.
- [65] Sir W. R. Hamilton. *Elements of Quaternions*, volume 1-2. Green and Co., Longmans, 1899.
- [66] Albert F. Harris, III and Robin Kravets. The design of a transport protocol for on-demand graphical rendering. In *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, NOSSDAV '02, pages 43–49, New York, NY, USA, 2002. ACM.
- [67] John C. Hart, George K. Francis, and Louis H. Kauffman. Visualizing quaternion rotation. *ACM Trans. Graph.*, 13:256–276, July 1994.
- [68] Frank Hartung, Uwe Horn, Jrg Huschke, Markus Kampmann, Thorsten Lohmar, and Magnus Lundevall. Delivery of broadcast services in 3G networks. *IEEE Transactions on Broadcasting*, 53(1):188 –199, Mar. 2007.
- [69] Ying He, Boon-Seng Chew, Dayong Wang, Chu-Hong Hoi, and Lap-Pui Chau. Streaming 3d meshes using spectral geometry images. In *Proceedings of the 17th ACM international conference on Multimedia*, MM '09, pages 431–440, 2009.
- [70] Ying He, Hongyu Wang, Chi-Wing Fu, and Hong Qin. A divide-and-conquer approach for automatic polycube map construction. *Computers and Graphics*, 33(3):369 – 380, 2009.
- [71] Hugues Hoppe. Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 99–108, New York, NY, USA, 1996. ACM.
- [72] Hugues Hoppe. Efficient implementation of progressive meshes. *Computers and Graphics*, 22(1):27–36, 1998.

- [73] Hugues Hoppe. Poisson surface reconstruction and its applications. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, SPM '08, pages 10–10, 2008.
- [74] D.A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098 –1101, Sept. 1952.
- [75] N. Hur, Hyun Lee, Gwang Soon Lee, Sang Jin Lee, A. Gotchev, and Sang-II Park. 3DTV Broadcasting and distribution systems. *IEEE Transactions on Broadcasting*, 57(2):395 –407, Jun. 2011.
- [76] M. Isenburg and P. Alliez. Compressing polygon mesh geometry with parallelogram prediction. In *Visualization, 2002. VIS 2002. IEEE*, pages 141 –146, Nov. 2002.
- [77] Zachy Karni and Craig Gotsman. Spectral compression of mesh geometry. In *Proc. SIGGRAPH'00*, pages 279–286, 2000.
- [78] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 61–70, 2006.
- [79] A. Khodakovsky, P. Alliez, M. Desbrun, and P. Schröder. Near optimal connectivity encoding of 2 manifold polygon meshes. *Graphical Models*, 64(3):147–168, 2002.
- [80] Andrei Khodakovsky, Peter Schröder, and Wim Sweldens. Progressive geometry compression. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 271–278, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [81] Donghyun Kim and Dongsuk Yook. Linear spectral transformation for robust speech recognition using maximum mutual information. *Signal Processing Letters, IEEE*, 14(7):496 –499, July 2007.

- [82] E. Lamboray, S. Wurmlin, and M. Gross. Real-time streaming of point-based 3d video. In *IEEE Proceedings of Virtual Reality, 2004.*, pages 91 – 281, March 2004.
- [83] P. Larochelle. Visualizing quaternions (a.j. hanson; 2006) [bookshelf]. *Control Systems, IEEE*, 28(4):104 –105, aug. 2008.
- [84] Bruno Lévy. Laplace-Beltrami eigenfunctions towards an algorithm that "understands" geometry. In *SMI*, page 13, 2006.
- [85] H. Li, M. Li, and B. Prabhakaran. Middleware for streaming 3d progressive meshes over lossy networks. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(4):282–317, 2006.
- [86] Weiping Li. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):301 –317, Mar. 2001.
- [87] Nein-Hsien Lin, Ting-Hao Huang, and Bing-Yu Chen. 3D Model streaming based on JPEG 2000. *IEEE Trans. Consumer Electron.*, 53(1):182–190, Feb. 2007.
- [88] Guodong Liu and Leonard McMillan. Segment-based human motion compression. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '06*, pages 127–135, 2006.
- [89] Guodong Liu, Jingdan Zhang, Wei Wang, and Leonard McMillan. Human motion estimation from a reduced marker set. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games, I3D '06*, pages 35–42, 2006.
- [90] Yang Liu, Balakrishnan Prabhakaran, and Xiaohu Guo. A robust spectral approach for blind watermarking of manifold surfaces. In *MM&Sec*, pages 43–52, 2008.

- [91] Yotam Livny, Soeren Pirk, Zhanglin Cheng, Feilong Yan, Oliver Deussen, Daniel Cohen-Or, and Baoquan Chen. Texture-lobes for tree modelling. *ACM Trans. Graph.*, 30:53:1–53:10, August 2011.
- [92] Stephane Mallat and Gabriel Peyre. A review of bandlet methods for geometrical image representation. *Numerical Algorithms*, 44:205–234, 2007.
- [93] Bruce Merry, Patrick Marais, and James Gain. Compression of dense and regular point clouds. In *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa, AFRIGRAPH '06*, pages 15–20, 2006.
- [94] Sebastien Mondet, Wei Cheng, Geraldine Morin, Romulus Grigoras, Frederic Boudon, and Wei Tsang Ooi. Streaming of plants in distributed virtual environments. In *Proc. ACM Multimedia*, pages 1–10, NY, USA, 2008.
- [95] Rachid Namane, Fatima O. Boumghar, and Kadi Bouatouch. QSPLAT compression. In *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa, AFRIGRAPH '04*, pages 15–24, 2004.
- [96] R. Pajarola and J. Rossignac. Compressed progressive meshes. *IEEE Trans. Vis. and Comput. Graphics*, 6(1):79–93, Jan-Mar 2000.
- [97] Sung-Bum Park, Chang-Su Kim, and Sang-Uk Lee. Progressive mesh compression using cosine index predictor and 2-stage geometry predictor. In *IEEE International Conference on Image Processing*, volume 2, pages 233–236, 2002.
- [98] Sung-Bum Park, Chang-Su Kim, and Sang-Uk Lee. Error resilient 3-d mesh compression. *IEEE Transactions on Multimedia*, 8(5):885–895, Oct. 2006.
- [99] Frédéric Payan and Marc Antonini. Temporal wavelet-based compression for 3d animated models. *Computers and Graphics*, 31(1):77–88, 2007.

- [100] Gabriel Peyré and Stéphane Mallat. Surface compression with geometric bandelets. In *Proc SIGGRAPH'05*, pages 601–608, NY, USA, 2005.
- [101] G. Popescu. On scheduling 3d model transmission in network virtual environments. In *Proceedings of the Sixth IEEE International Workshop on Distributed Simulation and Real-Time Applications*, DS-RT '02, pages 127–, Washington, DC, USA, 2002. IEEE Computer Society.
- [102] Jovan Popović and Hugues Hoppe. Progressive simplicial complexes. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 217–224, 1997.
- [103] M. Preda and F. Preteux. Insights into low-level avatar animation and MPEG-4 standardization. *Signal Processing: Image Communication*, 17(9):717–741, Oct 2002.
- [104] Marius Preda, Blagica Jovanova, Ivica Arsov, and Françoise Prêteux. Optimized MPEG-4 animation encoder for motion capture data. In *Proceedings of the twelfth international conference on 3D web technology*, Web3D '07, pages 181–190, 2007.
- [105] Przemyslaw Prusinkiewicz, Lars Mündermann, Radoslaw Karwowski, and Brendan Lane. The use of positional information in the modeling of plants. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 289–300, 2001.
- [106] Binjie Qin, Zhijun Gu, Xianjun Sun, and Yisong Lv. Registration of images with outliers using joint saliency map. *Signal Processing Letters, IEEE*, 17(1):91–94, Jan. 2010.
- [107] P. E. Hart R. O. Duda and D. G. Stork. *Pattern Classification*. John Wiley, New York, 2nd edition edition, 2001.

- [108] K. R. Rao and J. J Hwang. *Techniques and standards for image, video, and audio coding*. Prentic Hall, New Jersey, 1996.
- [109] K. R. Rao, D. N. Kim, and J. J Hwang. *Fast Fourier Transform: Algorithms, Advantages, Applications*. Springer, Heidelberg , Germany, 2010.
- [110] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, San Diego, CA, USA, 1990.
- [111] Patrick Reuter, Ireneusz Tobor, Christophe Schlick, and Sébastien Dedieu. Point-based modelling and rendering using radial basis functions. In *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, GRAPHITE '03, pages 111–118, 2003.
- [112] Guodong Rong, Yan Cao, and Xiaohu Guo. Spectral mesh deformation. *The Visual Computer*, 24(7-9):787–796, 2008.
- [113] Jarek Rossignac. Edgebreaker: Connectivity compression for triangle meshes. *IEEE Transactions on Visualization and Computer Graphics*, 5:47–61, 1999.
- [114] J. Roy, R. Balter, and C. Bouville. Hierarchical representation of virtual cities for progressive transmission over networks. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 432 –439, june 2006.
- [115] Szymon Rusinkiewicz and Marc Levoy. Streaming QSPLAT: a viewer for networked visualization of large, dense models. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, I3D '01, pages 63–68, 2001.
- [116] Raif M. Rustamov. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Symposium on Geometry Processing*, pages 225–233, 2007.

- [117] John Schlag. *Using Geometric Constructions to Interpolate Orientation with Quaternions, Graphics Gems II*. James Arvo Editor, Academic Press, 1991.
- [118] Dieter Schmalstieg and Michael Gervautz. Demand-driven geometry transmission for distributed virtual environments. In *Computer Graphics Forum*, pages 421–433, 1996.
- [119] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the h.264/avc standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103 –1120, sept. 2007.
- [120] S.D. Servetto and K. Nahrstedt. Video streaming over the public internet: multiple description codes and adaptive transport protocols. In *Proceedings of the International Conference on Image Processing, ICIP 99.*, volume 3, pages 85 –89 vol.3, 1999.
- [121] S.D. Servetto, K. Ramchandran, V.A. Vaishampayan, and K. Nahrstedt. Multiple description wavelet based image coding. *Image Processing, IEEE Transactions on*, 9(5):813 –826, may 2000.
- [122] S.D. Sevetto and K. Nahrstedt. Broadcast quality video over ip. *Multimedia, IEEE Transactions on*, 3(1):162 –173, mar 2001.
- [123] Shu Shi, Cheng-Hsin Hsu, Klara Nahrstedt, and Roy Campbell. Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming. In *Proceedings of the 19th ACM international conference on Multimedia*, MM ’11, pages 103–112, New York, NY, USA, 2011. ACM.
- [124] Shu Shi, Klara Nahrstedt, and Roy H. Campbell. View-dependent real-time 3d video compression for mobile devices. In *Proceedings of the 16th ACM international conference on Multimedia*, MM ’08, pages 781–784, New York, NY, USA, 2008. ACM.

- [125] Ken Shoemake. Animating rotation with quaternion curves. *SIGGRAPH Comput. Graph.*, 19:245–254, July 1985.
- [126] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph.*, 22:382–391, July 2003.
- [127] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered principal components for precomputed radiance transfer. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 382–391, 2003.
- [128] Olga Sorkine, Daniel Cohen-Or, and Sivan Toledo. High-pass quantization for mesh encoding. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '03, pages 42–51, Aire-la-Ville, Switzerland, 2003. Eurographics Association.
- [129] G. Taubin, W.P. Horn, F. Lazarus, and J. Rossignac. Geometry coding and vrml. *Proceedings of the IEEE*, 86(6):1228–1243, Jun 1998.
- [130] Gabriel Taubin. A signal processing approach to fair surface design. In *SIGGRAPH'95*, pages 351–358, 1995.
- [131] Gabriel Taubin, André Guézic, William Horn, and Francis Lazarus. Progressive forest split compression. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 123–132, 1998.
- [132] Gabriel Taubin and Jarek Rossignac. Geometric compression through topological surgery. *ACM TRANSACTIONS ON GRAPHICS*, 17:84–115, 1998.
- [133] D. S. Taubman and M. W. Marcellin. *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Kluwer Academic, Norwell, Massachusetts, USA, 2009.

- [134] Dihong Tian. *Streaming Three-Dimensional Graphics with Optimized Transmission and Rendering Scalability*. Phd Thesis, Georgia Institute of Technology, 2006.
- [135] Costa Touma and Craig Gotsman. Triangle mesh compression. In *Graphics Interface*, pages 26–34, 1998.
- [136] Yuh-Ren Tsai and Li-Cheng Lin. Sequential fusion for distributed detection over bsc channels in an inhomogeneous sensing environment. *Signal Processing Letters, IEEE*, 17(1):99 –102, Jan. 2010.
- [137] V.S. Tseng and Ching-Pin Kao. A novel similarity-based fuzzy clustering algorithm by integrating PCM and mountain method. *IEEE Transactions on Fuzzy Systems*, 15(6):1188 –1196, Dec. 2007.
- [138] Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. *Comput. Graph. Forum*, 27(2):251–260, 2008.
- [139] A. Vetro, A.M. Tourapis, K. Muller, and Tao Chen. 3d-tv content storage and transmission. *IEEE Transactions on Broadcasting*, 57(2):384 –394, June 2011.
- [140] Hongyu Wang, Ying He, Xin Li, Xianfeng Gu, and Hong Qin. Polycube splines. *Computer Aided Design*, 40(6):721 – 733, 2008.
- [141] Yu Wang and Lap-Pui Chau. Bit-rate allocation for broadcasting of scalable video over wireless network. , *IEEE Transactions on Broadcasting*, 56(3):288 –295, Sep. 2010.
- [142] Wanmin Wu, Ahsan Arefin, Gregorij Kurillo, Pooja Agarwal, Klara Nahrstedt, and Ruzena Bajcsy. Color-plus-depth level-of-detail in 3d tele-immersive video: a psychophysical approach. In *Proceedings of the 19th ACM international conference on Multimedia*, MM ’11, pages 13–22, New York, NY, USA, 2011. ACM.

- [143] Y. Xiang and A.T.L. Phuan. Feature extraction using the k-means fast learning artificial neural network. In *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia.*, volume 2, pages 1004 – 1008 vol.2, dec. 2003.
- [144] Rui Xu and Donald C. Wunsch II. *Clustering*. John Wiley, New Jersey, USA, 2008.
- [145] Sheng Yang, Chao-Hua Lee, and C.-C. Jay Kuo. Optimized mesh and texture multiplexing for progressive textured model transmission. In *Proc. ACM Multimedia*, pages 676–683, NY, USA, 2004.
- [146] Hao Zhang, Oliver van Kaick, and Ramsay Dyer. Spectral methods for mesh processing and analysis. In *Proc. of Eurographics State-of-the-art Report*, pages 1–22, 2007.
- [147] Liang Zhang, C. Vazquez, and S. Knorr. 3d-tv content creation: Automatic 2d-to-3d video conversion. *IEEE Transactions on Broadcasting*, 57(2):372–383, June 2011.
- [148] Denis Zorin, Peter Schröder, and Wim Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, SIGGRAPH '96*, pages 189–192, 1996.