Surface Motion Graphs for 3D Video-based Animation of People

Peng Huang

Submitted for the Degree of Doctor of Philosophy from the University of Surrey

Centre for Vision, Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, UK

November 2009

© Peng Huang 2009
Summary

Multiple view reconstruction of human performance as a 3D video has advanced to the stage of capturing detailed non-rigid dynamic surface shape and appearance of the body, clothing and hair during motion. Full 3D video scene capture holds the potential to create truly realistic animation by reproducing the dynamics of shape and appearance currently missing from marker-based motion capture. The acquisition results in an unstructured volumetric or mesh approximation of the surface shape at each time instance without temporal correspondence. This makes the reuse of 3D video data more challenging than conventional skeletal motion capture data. Previous research employed a directed graph structure to represent possible transitions between motion capture sequences. However, these approaches only deal with low degree-of-freedom (DOF) skeletal motion and cannot be directly extended to high DOF 3D video surface motion. In this thesis, we introduce a framework which automatically constructs motion graphs for 3D video sequences and synthesises novel animations to satisfy user-defined constraints on movement, location and timing.

The framework comprises two stages: pre-processing the database of 3D video sequences into a surface motion graph; and synthesis by optimising the graph-path to satisfy user-defined constraints and minimise the transition cost. The surface motion graph for a set of 3D videos represents the possible transitions between sequences and self-transitions within a sequence. This representation is analogous to motion graphs for motion capture sequences. Motion capture data has known temporal correspondence allowing the definition of similarity metrics to identify transitions that will not cause unnatural motion. Accurate estimation of temporal surface correspondence for 3D video sequences of human motion is still an open-problem. Current approaches are computationally expensive and require manual intervention to obtain correct correspondences for complex non-rigid surface motion. In this thesis, we introduce temporal shape similarity based on shape histograms to identify transitions without knowing temporal correspondence. The comprehensive evaluation of shape similarity metrics shows that this measure gives the best performance for identifying frames with similar shape and motion compared to other previously introduced shape similarity measures which do not require temporal correspondence. This approach greatly increase the flexibility in the reuse of 3D video sequences allowing specification of high-level user constraints to produce novel complex 3D video sequences of human motion. Results on real 3D video sequences of performers wearing a variety of clothing demonstrate that the proposed approach allows concatenative synthesis of novel sequences which preserve the realism of the captured 3D video.

Key words: Human Motion Synthesis, Surface Motion Graphs, 3D Video, 3D Character Animation, 3D Object Retrieval.

Email: p.huang@surrey.ac.uk
WWW: http://personal.ee.surrey.ac.uk/Personal/P.Huang/
Acknowledgements

Firstly, I would like to thank my supervisor, Prof. Adrian Hilton, for giving me the opportunity to undertake research in computer vision & graphics, and for his thoughtful guidance throughout. I would also like to thank Dr. Jonathan Starck, for his collaborated work and kindly help during my daily research in the first two years PhD study. Most ideas of this thesis come from the discussion between three of us. I am grateful to the head of centre, Prof. Josef Kittler and all my friends and colleagues at CVSSP. They are a group of talented and friendly people. Finally, I would like to thank my family for their love and support.

My PhD was financially supported by Video-based Animation of People (VBAP), an EPSRC project (EP/E001351/1) and CVSSP Oversea Studentship.
Contents

1 Introduction .............................................. 1
  1.1 Background ........................................ 3
  1.2 Problem Statement ................................ 4
  1.3 System Overview ................................ 5
  1.4 Contributions .................................. 6
  1.5 Publications ................................... 7
  1.6 Applications ................................... 8

2 Literature Review ....................................... 9
  2.1 Example-based Synthesis ............................ 10
    2.1.1 2D Video .................................... 10
    2.1.2 Motion Capture ................................ 13
    2.1.3 3D Video .................................... 17
  2.2 3D Shape Similarity ................................ 27
    2.2.1 Feature-based Methods ........................ 27
      2.2.1.1 Global Feature ............................ 28
      2.2.1.2 Local Feature ............................. 29
      2.2.1.3 Local Feature Distribution .............. 31
      2.2.1.4 Compact Canonical Representations ....... 34
    2.2.2 Graph-based Methods .......................... 35
      2.2.2.1 B-rep Graph ................................ 35
      2.2.2.2 Skeletal Graph ............................. 38
      2.2.2.3 Reeb Graph ................................ 41
    2.2.3 View-based Methods ........................... 43
    2.2.4 Bending-invariant Methods .................. 45
  2.3 Shape-Colour Similarity ............................ 46
  2.4 Temporal Similarity ................................ 48
  2.5 Self-Similarity Matrix ............................ 50
  2.6 Summary .......................................... 50

3 3D Shape Similarity .................................... 51
  3.1 3D Shape Descriptors ............................... 52
    3.1.1 Shape Distribution [91] ....................... 52
    3.1.2 Spin Image [58] ................................ 53
    3.1.3 Shape Histogram [3] ........................... 55
    3.1.4 Spherical Harmonics [62] ..................... 57
3.2 Static Similarity Metrics ........................................... 59
3.3 Performance Evaluation ........................................... 61
   3.3.1 Ground Truth .................................................. 61
   3.3.2 Evaluation Criterion ........................................ 65
   3.3.3 Parameter Setting ........................................... 66
      3.3.3.1 Shape Distribution .................................... 66
      3.3.3.2 Spin Image ............................................. 66
      3.3.3.3 Shape Histogram ........................................ 66
      3.3.3.4 Spherical Harmonics .................................... 67
   3.3.4 Evaluation of 3D Shape Descriptors ................. 67
3.4 Similarity Measure on Real Data .............................. 69
3.5 Summary ................................................................. 70

4 Temporal Similarity .................................................. 81
   4.1 Temporal Descriptors and Similarity Metrics ............ 81
      4.1.1 Time-filtered Descriptors ............................... 83
      4.1.2 Shape-flow Descriptors .................................. 83
   4.2 Performance Evaluation ......................................... 87
      4.2.1 Evaluation of time-filtered descriptors ............... 88
      4.2.2 Evaluation of shape-flow descriptors .................. 90
   4.3 Similarity Measure on Real Data ............................ 92
   4.4 Summary ................................................................. 94

5 Shape-Colour Similarity ............................................... 99
   5.1 Shape-Colour Descriptors ...................................... 99
      5.1.1 Shape-Colour Histograms ................................ 100
      5.1.2 Descriptors for Comparison .............................. 103
   5.2 Similarity Metrics ................................................ 104
   5.3 Performance Evaluation ......................................... 105
      5.3.1 Ground Truth ............................................... 105
      5.3.2 Evaluation of Shape-Colour Descriptors ............... 108
   5.4 Similarity Measure on Real Data ............................ 109
   5.5 Discussion ............................................................. 111
   5.6 Summary ................................................................. 113

6 3D Video Key-Frame Extraction .................................... 115
   6.1 Background .......................................................... 116
   6.2 Similarity-based Key-frame Extraction ..................... 118
      6.2.1 Rate and Distortion ....................................... 118
      6.2.2 Conciseness Cost Matrix .................................. 119
      6.2.3 Graph Construction ......................................... 119
      6.2.4 Shortest Path ............................................... 121
      6.2.5 Optimal Point on the R-D curve ......................... 122
   6.3 Experimental Results and Evaluation ...................... 122
   6.4 Discussion ............................................................. 123
   6.5 Summary ................................................................. 125
Contents

7 Surface Motion Graphs 129
  7.1 Building a Surface Motion Graph .................................. 130
    7.1.1 Transitions ..................................................... 130
      7.1.1.1 Static Similarity Measure .............................. 131
      7.1.1.2 Adaptive Temporal Filtering ......................... 132
    7.1.2 Graph Construction ........................................... 134
  7.2 Motion Synthesis .................................................... 134
    7.2.1 Cost Function .................................................. 136
    7.2.2 Path Optimisation .............................................. 137
      7.2.2.1 Graph Path Decomposition ............................. 139
      7.2.2.2 Integer Linear Programming ............................ 139
      7.2.2.3 Graph Path Composition ............................... 140
  7.3 Experimental Results and Evaluation .............................. 141
  7.4 Discussion ............................................................ 142
  7.5 Summary ............................................................... 144

8 Conclusions 151
  8.1 Contributions ....................................................... 152
  8.2 Limitations ........................................................... 154
  8.3 Future Work .......................................................... 155

A ROC performance for Static and Temporal Shape Descriptors 157

B Algorithms for Surface Motion Graphs 161
  B.1 Find All Walks and Loops by DFS ................................ 162
  B.2 Feedback Strategy .................................................. 162

Bibliography 163
## List of Figures

1.1 Example frames of a 3D video .............................................. 2
1.2 The Motivation ................................................................. 3
1.3 System overview ............................................................... 5

2.1 Video Rewrite [13] .............................................................. 10
2.2 Video Sprites [95] .............................................................. 12
2.3 Transition [105] ................................................................. 14
2.4 Two-layer Graph-like Structure [67] ..................................... 15
2.5 Motion Annotations [5] .......................................................... 15
2.6 Motion Graphs [64] .............................................................. 16
2.7 Surface Motion Capture [102] .............................................. 20
2.8 Wide-timeframe Surface Correspondence [101] ...................... 21
2.9 Surface Correspondence with topological changes [112] .......... 23
2.10 Performance capture [1] ....................................................... 25
2.11 Surface Motion Synthesis [103] ........................................... 26
2.12 Concatenative Motion Editing [119] ................................... 27
2.13 Reflective Symmetry [61] ...................................................... 28
2.14 Spin Image [58] ................................................................. 31
2.15 Compound Spin Image [7] .................................................... 32
2.16 Shape Distribution [91] ....................................................... 33
2.17 Shape Histogram [3] ......................................................... 34
2.18 Shape Context [9] ............................................................... 35
2.19 Spherical Harmonics Representation [36] ............................ 36
2.20 Spherical Harmonics Representation Ambiguity [62] ............. 37
2.21 B-rep Graph [30] ............................................................... 37
2.22 Model Signature Graph (MSG) [81] ..................................... 38
2.23 Skeleton Matching [104] ..................................................... 39
2.24 Skeletal Graph [56] ............................................................ 40
2.25 Skeleton Recovery [82] ....................................................... 41
2.26 Multi-resolutional Reeb Graph (MRG) [43] .......................... 42
2.27 Augmented Multi-resolution Reeb Graph (aMRG) [110] ....... 42
2.28 Light Field [19] ................................................................. 44
2.29 Bending-invariant Descriptor [31] ...................................... 46
2.30 Space-Time Shapes [40] ...................................................... 49

3.1 Illustration of Shape Distribution ......................................... 54
3.2 Illustration of Spin Image .................................................... 55
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>Illustration of Shape Histogram</td>
<td>57</td>
</tr>
<tr>
<td>3.4</td>
<td>Illustration of Spherical Harmonic representation (SHR)</td>
<td>59</td>
</tr>
<tr>
<td>3.5</td>
<td>Synthetic Dataset</td>
<td>62</td>
</tr>
<tr>
<td>3.6</td>
<td>Ground-Truth</td>
<td>64</td>
</tr>
<tr>
<td>3.7</td>
<td>Parameter Setting</td>
<td>72</td>
</tr>
<tr>
<td>3.8</td>
<td>Comparison for Shape Descriptors</td>
<td>73</td>
</tr>
<tr>
<td>3.9</td>
<td>Static Similarity Measure Ambiguity</td>
<td>74</td>
</tr>
<tr>
<td>3.10</td>
<td>Real Dataset Example Motions</td>
<td>75</td>
</tr>
<tr>
<td>3.11</td>
<td>[Self-Similarity for Real Dataset]</td>
<td>76</td>
</tr>
<tr>
<td>3.11</td>
<td>Self-Similarity for Real Dataset</td>
<td>77</td>
</tr>
<tr>
<td>3.12</td>
<td>Cross-Similarity for Real Dataset</td>
<td>78</td>
</tr>
<tr>
<td>3.13</td>
<td>Similarity Measure on Real Data</td>
<td>79</td>
</tr>
<tr>
<td>4.1</td>
<td>Temporal Similarity Measure against increasing window size</td>
<td>82</td>
</tr>
<tr>
<td>4.2</td>
<td>Shape-Flow Matching</td>
<td>85</td>
</tr>
<tr>
<td>4.3</td>
<td>Temporal Similarity Measure Comparison</td>
<td>86</td>
</tr>
<tr>
<td>4.4</td>
<td>SHvrS Failure</td>
<td>88</td>
</tr>
<tr>
<td>4.5</td>
<td>Combined ROC for Static and Filtered descriptors</td>
<td>89</td>
</tr>
<tr>
<td>4.6</td>
<td>Combined ROC for Shape-Flow descriptors</td>
<td>91</td>
</tr>
<tr>
<td>4.7</td>
<td>Intra-person Similarity for Real Data</td>
<td>96</td>
</tr>
<tr>
<td>4.7</td>
<td>Intra-person Similarity for Real Data (continued)</td>
<td>97</td>
</tr>
<tr>
<td>4.7</td>
<td>Intra-person Similarity for Real Data (continued)</td>
<td>98</td>
</tr>
<tr>
<td>4.8</td>
<td>Inter-person Similarity for Real Data</td>
<td>98</td>
</tr>
<tr>
<td>5.1</td>
<td>Illustration of Shape-Colour Histogram</td>
<td>102</td>
</tr>
<tr>
<td>5.2</td>
<td>Cameras and Lighting for simulated Data</td>
<td>106</td>
</tr>
<tr>
<td>5.3</td>
<td>Example frames of synthetic data</td>
<td>107</td>
</tr>
<tr>
<td>5.4</td>
<td>Ground-Truth Similarity Matrix</td>
<td>108</td>
</tr>
<tr>
<td>5.5</td>
<td>ROC for Shape-Colour and Shape-Only Descriptors</td>
<td>109</td>
</tr>
<tr>
<td>5.6</td>
<td>Comparison for Temporal Shape-Colour and Shape-Only Descriptors</td>
<td>110</td>
</tr>
<tr>
<td>5.7</td>
<td>Shape-Colour Similarity Measure for Real Data</td>
<td>112</td>
</tr>
<tr>
<td>6.1</td>
<td>Conciseness Matrix</td>
<td>120</td>
</tr>
<tr>
<td>6.2</td>
<td>Graph Construction from Conciseness Matrix</td>
<td>121</td>
</tr>
<tr>
<td>6.3</td>
<td>Key-Frame Extraction is solved as Shortest Path</td>
<td>122</td>
</tr>
<tr>
<td>6.4</td>
<td>Self-Similarity based Key-Frame Extraction vs. Baseline</td>
<td>124</td>
</tr>
<tr>
<td>6.5</td>
<td>Key-Frame Extraction Results (More)</td>
<td>125</td>
</tr>
<tr>
<td>6.6</td>
<td>R-D curves for multiple 3D video</td>
<td>127</td>
</tr>
<tr>
<td>7.1</td>
<td>User-defined Constraints</td>
<td>130</td>
</tr>
<tr>
<td>7.2</td>
<td>Illustration of a transition</td>
<td>131</td>
</tr>
<tr>
<td>7.3</td>
<td>Weighting functions</td>
<td>132</td>
</tr>
<tr>
<td>7.4</td>
<td>Surface Motion Graph</td>
<td>135</td>
</tr>
<tr>
<td>7.5</td>
<td>Graph Decomposition</td>
<td>139</td>
</tr>
<tr>
<td>7.6</td>
<td>Motion Synthesis Results</td>
<td>143</td>
</tr>
<tr>
<td>7.7</td>
<td>Evaluation for Character 1: Stand#1 → Walk2Stand30#30</td>
<td>146</td>
</tr>
<tr>
<td>7.8</td>
<td>Evaluation for Character 1: Stand#1 → Hit#45</td>
<td>147</td>
</tr>
<tr>
<td>7.9</td>
<td>Evaluation for Fashion 1: Pose#1 → Twirl#85</td>
<td>148</td>
</tr>
</tbody>
</table>
List of Figures

7.10 Evaluation for Fashion 2: Pose#1 → Twirl#100 .......................... 149
A.1 ROC for Static Descriptors .................................................. 158
A.2 ROC for Filtered Descriptors ................................................. 159
A.3 ROC for Shape-Flow Descriptors ........................................... 160
List of Tables

3.1 Optimal Parameter Setting for Shape Descriptors .................. 68
4.1 Synthetic 3D Video Database ........................................ 90
4.2 Computational Cost against Window Size .......................... 92
4.3 Real 3D Video Database ............................................. 93
7.1 Quantitative Evaluation of Synthesis Results ..................... 145
List of Variables

- **P**
  A 3D video database

- **N_P**
  Number of sequences in P

- **P, Q**
  A 3D video sequence

- **N_P, N_Q**
  Number of frames in P and Q

- **p, q**
  A 3D video frame

- **M =< V, F >**
  A 3D triangle mesh

  - **V, F**
    A set of vertices and a set of triangles

  - **N_V, N_F**
    Number of vertices and triangles in V and F

- **v = (x, y, z)**
  3D position of a vertex in Cartesian coordinate system

- **f =< v_0, v_1, v_2 >**
  A face of a triangle mesh

- **I**
  A set of multiple-view images for a 3D video frame

- **N_I**
  Number of frames for I

- **I**
  A single-view image in I

- **d**
  Euclidean distance between a pair of random points on surface

- **Δd**
  Bin size for d

- **N_p, N_d**
  Number of samples (point-pairs) and fixed sized bins

- **α, β**
  The distance along and from the principal axis

- **Δα, Δβ**
  Bin size for α and β

- **N_α, N_β**
  Number of bins for α and β

- **s = (r, θ, φ)**
  3D position of a point in Spherical coordinate system

- **Δr, Δθ, Δφ**
  Bin size for r, θ and φ

- **N_r, N_θ, N_φ**
  Number of bins for r, θ and φ

- **f(r, θ, φ)**
  Spherical function derived from a 3D model

- **f_r(θ, φ)**
  Spherical function at radii r

- **f_r^l(θ, φ)**
  Spherical function at radii r of degree l

- **Y_r^m(θ, φ)**
  Spherical basis function of degree l and order m

- **N_R, N_L**
  Number of radial shells and frequency component preserved

- **S, S^t**
  Static and temporal similarity matrix

- **N_t**
  Define a temporal window size that equals to 2N_t + 1

- **s_{ij}, s_{ij}^t**
  Dissimilarity at (i, j) of static and temporal similarity matrix
\begin{tabular}{ll}
\hline
$\tau$ & Binary classification threshold \\
$C(\tau)$ & Binary classification matrix subject to a threshold $\tau$ \\
$c_{ij}(\tau)$ & Binary classification at $(i,j)$ subject to a threshold $\tau$ \\
$SGT$ & Ground-truth similarity matrix \\
$s_{ij}^{GT}$ & Ground-truth similarity score at $(i,j)$ of a similarity matrix \\
$\tau^{GT}$ & Ground-truth binary classification threshold \\
$C^{GT}(\tau)$ & Ground-truth binary classification matrix subject to a threshold $\tau$ \\
$c_{ij}^{GT}(\tau)$ & Ground-truth binary classification at $(i,j)$ subject to a threshold $\tau$ \\
$ts(\tau), fs(\tau)$ & Counts of true-similar and false-similar predictions subject to $\tau$ \\
$td(\tau), fd(\tau)$ & Counts of true-dissimilar and false-dissimilar predictions subject to $\tau$ \\
$TPR(\tau), FPR(\tau)$ & True-Positive and False-Positive Rate subject to $\tau$ \\
c & $=(R, G, B)$ & Colour of a point in the RGB colour space \\
$\Delta R, \Delta G, \Delta B$ & Bin size for colour component $R, G$ and $B$ \\
$N_{R}, N_{G}, N_{B}$ & Number of bins for $R, G$ and $B$ \\
$\epsilon$ & Parameter for trade-off between Rate and Distortion \\
$K(\epsilon)$ & Key-frame selection subject to $\epsilon$ \\
$K^{opt}_{\epsilon}$ & Optimal key-frame selection subject to $\epsilon$ \\
$R(K)$ & Rate/Compactness subject to key-frame selection $K$ \\
$D(K)$ & Distortion/Faithfulness subject to key-frame selection $K$ \\
$C(K)$ & Conciseness Cost subject to key-frame selection $K$ \\
$T$ & A transition sequence \\
t & A transition frame \\
$m, n, N_{T}$ & Define a transition window: \\
m & Index of starting frame in "from" sequence \\
n & Index of starting frame in "to" sequence \\
$N_{T}$ & Number of frames in $T$ \\
d(T) & Distortion of transition $T$ \\
s(T) & Temporal filtered similarity of transition $T$ \\
$\chi(k)$ & Weight function for the temporal filtering \\
$T^{opt}$ & Optimal transition sequence \\
$m^{opt}, n^{opt}, N_{T}^{opt}$ & Define an optimal transition window: \\
dv, tv & User-specified target distance and time \\
F & A graph path on the Surface Motion Graph \\
C(F) & Combined cost of $F$ \\
$C_{s}(F), C_{d}(F), C_{t}(F)$ & Smoothness, location and time cost of $F$ \\
w_{d}, u_{t} & Weight for combining $C_{d}(F)$ and $C_{t}(F)$ \\
$F = L \cdot n$ & Define a composition of a graph path \\
\hline
\end{tabular}
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>A collection of a walk and attached loops</td>
</tr>
<tr>
<td>( l_0 )</td>
<td>A walk (graph path without any loop)</td>
</tr>
<tr>
<td>( l )</td>
<td>A graph path (either a walk or a loop)</td>
</tr>
<tr>
<td>( f_d(l) )</td>
<td>Traverse distance cost for ( l )</td>
</tr>
<tr>
<td>( f_t(l) )</td>
<td>Traverse time cost for ( l )</td>
</tr>
<tr>
<td>( f_s(l) )</td>
<td>Smoothness cost for ( l )</td>
</tr>
<tr>
<td>( n )</td>
<td>Number of repetitions correspond to ( l )</td>
</tr>
<tr>
<td>( n^{opt} )</td>
<td>Optimal number of repetitions correspond to ( L )</td>
</tr>
<tr>
<td>( F^{opt} )</td>
<td>Optimal graph path subject to user-defined constraints</td>
</tr>
</tbody>
</table>
List of Tables
Chapter 1

Introduction

Computer generated three-dimensional (3D) graphics has been widely used for the production of visual contents in the film, television and game industries. One of the key challenges for 3D content production is the creation of a realistic performance of people. Traditionally, such a performance is manually created by skilled artists and animators: artists sketch a character model and animators make it move. The whole process is time-consuming, labour-intensive and high cost. Moreover, the produced animation results often appear synthetic when compared to the real world.

Motion Capture techniques help create a more believable motion in less time with less labour involved by capturing skeletal motions of people. Skeletal motion capture of a person is performed by acquiring the 3D position of a sparse set of markers and estimating the corresponding skeletal pose. To produce a character animation, Motion Capture data is transferred onto a pre-built character model. However, building such a model and transferring motion still requires expert knowledge. Moreover, Motion Capture cannot capture non-skeleton motion like surface dynamics, for example, it cannot capture wrinkles in cloth or hair motion, therefore losing important details of a real vivid performance.

In recent decades, 3D video capture techniques have been introduced to capture shape, appearance and motion of the human body from multiple-view video from the real world. 3D geometry (shape) can be reconstructed with detailed surface dynamics and the appearance (colour) can be projected back to achieve video-quality realism. In addition, building a character model or transferring motion is not required. Therefore,
Chapter 1. Introduction

Figure 1.1: Example frames of a 3D video. Rendering from a virtual camera view: (a) 3D mesh model with texture mapping; (b) 3D mesh model.

3D video holds the potential to create a highly realistic character animation of human performance with less time and less labour involved. A typical 3D video contains a sequence of reconstructed 3D meshes with each mesh associated with a set of original captured multiple-view images which allows realistic texture mapping. Figure 1.1 shows some example frames from a 3D video with and without texture mapping. Since the user is free to control the camera while viewing the result, 3D video is also known as free-viewpoint video. However, current techniques can only replay a 3D video which limits the application. More interesting and useful work like 3D video-based human motion synthesis has been performed only with a lot of manual work involved [103, 119]. The challenge addressed in this thesis is to automatically or semi-automatically reuse captured 3D videos for human motion synthesis and create a more realistic character animation with less time and labour.

Motion synthesis on Motion Capture have been researched for many years. There are mainly two categories: learning-based and example-based methods. Learning-based methods, deriving a statistical motion model to describe the general motion characteristics and synthesise motion from the estimated motion model. This approach cannot guarantee that the synthesised result is physically realistic or looks natural. A large
1.1. Background

The reuse of captured temporal sequences of people (2D video, MoCap, 3D video) for animation production is an important problem because people form a central focus of most entertainment production (film, TV, game). To create a novel 2D video from captured ones, example-based methods such as Video-based Animation techniques [96] amount of training data is required which is now available for Motion Capture but not for 3D video. Example-based methods which concatenate motion segments provide an attractive alternative as there is no loss of detail from the original motion dynamics and no requirement for extra training data. Example-based synthesis for Motion Capture has advanced to allow user’s high-level controls on animated character’s motion, position and timing, which increases the capability and flexibility of the Motion Capture-based character animation system. Since they all construct a directed graph or a graph-like structure, they are referred to as Motion Graphs techniques.

This thesis introduces a framework of 3D video-based character animation, termed Surface Motion Graphs, as an analog of Motion Graphs. Highly realistic human motion could be synthesised by concatenating captured 3D videos whilst allowing user-defined high-level constraints on motion, position and timing. Figure 1.2 illustrates the motivation and the question mark shows how Surface Motion Graphs fit into the development of realistic character animation production.

Figure 1.2: The motivation to reuse 3D video.
employs a directed graph to represent transitions between 2D videos and search for a path over a graph satisfying user constraints. Similarly, to create a novel skeletal motion from Motion Capture (MoCap) data, Motion Graphs techniques [105, 67, 64, 4] employs a directed graph to represent transitions between Motion Capture sequences and search for a path on graph satisfying user constraints. For 3D video, there are only manual approaches [102, 119] available. An automatic or semi-automatic solution with high-level user control is highly desired. To be analogous to Video-based Animation techniques for 2D video and Motion Graphs techniques for MoCap, we introduce Surface Motion Graphs for 3D video.

1.2 Problem Statement

Given a set of 3D video sequences of a person performing different movements, the problem is to create novel animations according to user input while maintaining the natural captured movement. This requires solving four key problems:

- Identify transitions between 3D videos which do not cause unnatural artifacts.
- Construct a graph connecting different 3D video sequences of people.
- Optimise a path through the graph to satisfy user-defined high-level constraints on the animation.
- Seamlessly concatenate 3D videos on the graph path to produce a realistic character animation.

Furthermore, identifying transitions requires a similarity metric to identify frames of the 3D video sequence with similar surface shape, appearance and motion. We define a 3D video as an unstructured mesh sequence. Each frame is a mesh approximation of the surface shape. There is no temporal correspondence, and reliably achieving accurate dense correspondence of dynamic surfaces still remains an open research problem [100, 101, 2, 1, 114, 112, 120]. Therefore, we cannot simply derive a similarity metrics from the temporal correspondence of surface point it is unknown.
1.3 System Overview

An overview of the approach introduced to solve the problem of animation from 3D video is shown in Figure 1.3. A 3D video database is first used to build a Surface Motion Graph (SMG) by automatically identifying transition points between sequences. Path optimisation is then performed to satisfy user-defined constraints. Once the optimal path has been found, the desired animation is synthesised by concatenating 3D videos of the path. To build a SMG, similarity is measured for different motions, either shape similarity in Chapter 3 or shape-colour similarity in Chapter 5; temporal similarity is introduced to preserve motion dynamics in Chapter 4 and then used to derive an adaptive temporal filter for identifying optimal transitions in Chapter 7. Path optimisation to satisfy user constraints on motion based on Integer Linear Programming (ILP) is also introduced in Chapter 7. Chapter 6 introduces a self-similarity-based 3D video key-frame extraction which provide a summarization of 3D video sequences allowing the user to browse the 3D videos and select key-frames. User-selected key-frames and user-specified total traverse distance and total traverse time together provide powerful and flexible high-level user control. Note that although building a SMG is time-consuming, once built, motion can be synthesised many times according to different constraints which is fast and has the potential to allow real-time interactive user input.
1.4 Contributions

This thesis presents research which has investigated a number of problems in the analysis and manipulation of 3D video in order to achieve animation from 3D video.

A summary of main contributions is provided below:

- In order to identify optimal transitions, similarity metrics including shape-only, shape-colour and their combination with temporal similarity are investigated. All of these methods do not require solving temporal surface correspondence for 3D video. (Chapter 3,4,5)

- Two novel shape-flow descriptors, Shape-Flow Histograms, are introduced to match frames of time-varying 3D mesh sequences of people with similar shape and motion without a prior model or temporal correspondence. (Chapter 4)

- A novel shape-colour descriptor, Shape-Colour Histograms, is introduced to match frames of 3D video sequences with similar shape and appearance extending previous shape histograms. (Chapter 5)

- A quantitative evaluation is presented to compare the performance of conventional shape descriptors, shape-flow descriptors and shape-colour descriptors against a synthetic ground-truth dataset of temporal surface sequences of people in which surface correspondence is predefined. This demonstrates that temporal shape histograms give the best performance for 3D video sequences of people. (Chapter 3,4,5)

- A self-similarity-based key-frame selection method for video summarization is introduced. Evaluation against previous key-frame selection methods demonstrates improved performance in summarising temporal surface sequences of people. (Chapter 6)

- An analog of Motion Graphs techniques for Motion Capture, Surface Motion Graphs for 3D video, is introduced to represent possible transitions between different motions and within the same motion without unnatural movement artifacts. (Chapter 7)
• An ILP-based graph path optimisation is introduced to synthesise desired animated contents according to user-defined constraints and maximise the transition smoothness. This achieves global optimal solution rather than local optimal solution that, in general, previous approaches did. (Chapter 7)

• A quantitative evaluation of Surface Motion Graphs on human motion synthesis and path optimisation from real 3D video sequences is presented. This demonstrates that Surface Motion Graphs allows flexible production of novel human motion which preserves the detailed dynamics of the original 3D video sequences and accurately satisfies user-defined constraints. (Chapter 7)

1.5 Publications

This research has resulted in several publications in the field of 3D video, 3D object retrieval, computer graphics and computer vision.

Publications resulting from this are:

• P. Huang, J. Starck and A. Hilton. A Study of Shape Similarity for Temporal Surface Sequences of People. In Proceedings of the Sixth International Conference on 3D Digital Imaging and Modeling (3DIM'07), pages 408-418, Montréal, Québec, Canada, August 2007. (Chapter 3)


• P. Huang, A. Hilton and J. Starck. Human Motion Synthesis from 3D Video. In the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2009 (CVPR'09), Miami, FL, USA, June 2009. (Chapter 7)
Chapter 1. Introduction

- P. Huang, A. Hilton and J. Starck. Shape Similarity for 3D Video Sequences of People. Accepted for publication in the International Journal of Computer Vision (IJCV) special issue on 3D Object Retrieval, August 2009. (Chapter 3,4)

- P. Huang and A. Hilton. Surface Motion Graphs for Character Animation from 3D Video. In the 36th International Conference and Exhibition on Computer Graphics and Interactive Techniques (ACM SIGGRAPH'09) Talk Session, New Orleans, LA, USA, August 2009. (Chapter 7)

- P. Huang and A. Hilton. Shape-Colour Histograms for Matching 3D Video Sequences. In the Twelfth IEEE International Conference on Computer Vision (ICCV'09) workshop on 3-D Imaging and Modeling (3DIM'09), Kyoto, Japan, September 2009. (Chapter 5)

1.6 Applications

There is a great interest to synthesize convincing looking human motion in the game and film industry. In a game, both player-directed human characters and non-player human characters are engaged in a variety of activities. Ideally, these figures look and behave like a human, the motion is smooth and natural. Although we have not achieved interactive control over a game character in the animation system, it can be incorporated in future. The interactive control of a truly realistic avatar is an interesting and prospective application in game industry. In a film, traditionally, less interest relays in computational motion synthesis largely because actors are still the best way to get high quality motion. This trend appears to be changing as computer graphics can provide more realistic imagery at interactive rates. 3D video and manipulation of 3D video provide more freedom to the production of a film. Amounts of time and cost could be moved and reduced from production to post-production. For example, the decision of the camera position/viewpoint could be postponed to post-production; the character motion could be manipulated in post-production, which is particularly useful, since the change of scene could be done without actor intervention. This thesis shows the potential to produce user-controlled truely realistic 3D character animation.
Chapter 2

Literature Review

In this chapter, related research in example-based synthesis from captured sequences including 2D video, Motion Capture and 3D video are first reviewed. The visual quality of synthesised results largely depends on transitions, the key problem is to find smooth transitions where different sequences can be concatenated seamlessly. To accurately locate transitions requires effective similarity metrics. As mentioned in the previous Chapter, this thesis focuses on the reuse of 3D video reconstructed by Surface Motion Capture (SurfCap) [102]: each frame contains a set of multiple-view images and an independently reconstructed 3D mesh, where the temporal correspondence is unknown. Due to the difficulty in estimating accurate temporal correspondence in 3D video, which remains an open problem, we investigate similarity metrics for shape, motion and appearance which do not require correspondence. Although the set of multiple-view images could be used to derive similarity metrics alone. This would be view dependent and sensitive to occlusion. Similarity metrics are therefore investigated based on the reconstructed 3D shape which avoid problems of occlusion and viewpoint change. Similarity metrics previously introduced for content-based 3D object retrieval which consider static geometric similarity (shape) are reviewed, to be more distinctive, their combination with temporal similarity (motion) and photometric similarity (appearance) are also reviewed.
2.1 Example-based Synthesis

Example-based synthesis techniques were introduced in speech to allow re-production of natural speech from a corpus of recorded speech data. Subsequently example-based approaches were exploited in computer graphics to reuse and modify video sequence. Example-based Synthesis on 2D video, Motion Capture and 3D video are reviewed in following sections.

2.1.1 2D Video

Bregler et al. [13] introduced video rewrite to create a novel video of a person speaking with lip and jaw movements that synchronise to the input novel audio track. This technique segments and labels the audio track and applies them to the video clip. Synthesis is performed by retrieving the mouth images of the training database, reordering them according to the input novel audio track and finally blending them into a background scene which contains natural head and eye movement. Figure 2.1 illustrates this process.

Similarly, Ezzat et al. [32] provide an audio-driven visual-speech animation system for a human subject. They construct Multidimensional Morphable Models (MMM)
2.1. Example-based Synthesis

to parametrise mouth images associated with human speech by shape and appearance parameters. The MMM is capable to compute mouth images given shape-appearance configuration and vice versa. The visual-speech synthesis is to map from an input phone stream which is pre-extracted from an input audio to a trajectory of parameters in the MMM space. This trajectory is used to synthesise an output mouth image sequences which is then composited into a background scene.

Both of their approaches focus on the synthesis of a human speech, in which case, the video has a strong association with the audio. This limits their application for synthesising a more common human motion which normally does not has the required audio clue. Additionally, for 2D video without the knowledge of 3D geometry, re-compositing synthesised mouth sequences into background sequences is difficult, especially when there are large changes in head pose, lighting and viewpoint.

Schödl et al. [96] introduce video textures to create a similar looking video from a source video by rearranging original frames. The source video is first represented as a Markov process with each state corresponding to a single frame and the probabilities corresponding to the likelihood of transitions from one frame to another. To make video textures look smooth, frames at transitions need to be similar both in appearance and motion. A temporal similarity metrics is then used to compute frame-to-frame image similarity over a window of frames. They also introduce video-based animation as an extension, in which the synthesis of video texture can be guided by a user through high-level interactive controls.

Schödl and Essa [95] extend this work to create character animations of a moving object which is called video sprites. They allow a user to specify animations using a cost function and provide an optimisation algorithm to minimise it. The optimisation is performed based on repeated partial replacements of the sequence which is similar to the dynamic programming. This approach is demonstrated by creating character animations of animals following a user-specified path. Figure 2.2 illustrates the process of rearranging individual frames to create a novel animation.

Their approaches may fail when the video contains a highly structured phenomena like full-body human motion, unless using some higher-level motion and structure analysis. However, to consistently recover the structure of human motion is very difficult,
especially for 2D video when there are occlusions and view point changes. Additionally, their optimisation algorithm gives the local optimal rather than global one. In this thesis, the proposed animation system is based on 3D video which can inherently handle both occlusions and view point changes. The proposed optimisation algorithm gives the global optimal according to user-defined constraints in movement, position and timing.

Recently, Flagg et al. [33] have exploited both 2D video and Motion Capture to generate controllable photorealistic animations of human performance. 2D video and MoCap data are first synchronised. A video graph is then constructed to represent transitions between motion segments. To identify transitions, they provide a similarity metric based on the 3D markers directly without a fitted skeleton. They compute the Euclidean distance of 3D marker positions and trajectories from frame to frame over a fixed time window. The combination with MoCap data gives accurate and reliable correspondence which is then used to blend images to create seamless transitions. However, their approach requires a human subject to wear tight clothing attached with markers performing movements in a style like a side-scrolling video game. Their approach is also sensitive to parallax effects (parallax can be observed, for example, in the relative motion of the shoulders with respect to the centre of the chest). This is because using a fixed single-view video they cannot deal with occlusion or view point change. In this thesis, there is no requirement on the captured human performance in terms of the costume, markers or motion style and the synthesised motion is not sensitive to occlusion or view point change.
In the next section, the review will focus on synthesising novel human skeletal motion by concatenating existing Motion Capture sequences.

2.1.2 Motion Capture

Tanco and Hilton [105] introduced a two-level statistical model to synthesise novel motions from a database of Motion Capture data. The similarity metrics based on joint rotations of a skeletal human model are defined. Similar frames are divided into clusters (states). Motion examples intersect with these clusters resulting motion segments. Their approach uses a Markov chain to represent transitions between states and a Hidden Markov Model to represent transitions between motion segments. The transition probability between two states is estimated by counting the frequency of consecutive frames occurring. The transition probability between two motion segments is chosen to discourage jumping between motion examples. According to user-defined start and end key-frames, initial and final states are chosen. The first stage synthesis is then performed by searching for an optimal state sequence which minimises the transition cost between states. The second stage synthesis takes the generated state sequence as an input and solves for most likely sequences of motion segments. Smooth transition are produced by blending two motion segments over a small interval of frames around the point where the two motion segments are closest. Figure 2.3 shows an example of a transition concatenating sitting and running. This approach demonstrates the synthesis of novel skeletal motions from a Motion Capture database. However, the similarity metric requires temporal correspondence of skeletal motion and it only measures skeletal pose similarity. This limits the extension to 3D video where the correspondence is unknown and surface similarity is required to produce smooth transitions. In addition, the user cannot control either character location or timing of motions between key-frames which are essential in the production of animation.

Lee et al. [67] provided a two-layer graph-like structure encoding possible links between different frames which allows efficient search and interactive control. The logic is that if two frames are sufficiently similar, their futures could be interchanged. The similarity metric based on the skeletal pose measures the difference in joint angles and velocities at various points across the body. The recursive search terminates when the depth of the
A spanning tree reaches a given maximum. Choice-based, sketch-based and performance-based interface are demonstrated for interactive control of character motion. Figure 2.4 shows an example of such a two-layer graph-like structure. Their similarity metric also requires temporal correspondence of skeletal motion and measure skeletal pose similarity only. Their path optimisation cannot guarantee to find global solution while the approach presented in this thesis is capable to give the global optimal path.

Similarly, Arikan and Forsyth [4] employed a direct graph to connect motion segments where each node corresponds to a motion sequence and each edge a transition. Their similarity metric measures the distance between two frames in terms of the differences between corresponding joint positions and velocities and the difference between the whole body velocities and accelerations. Hard and soft constraints, e.g. the length of the motion, the body's position and orientation at a particular frame are defined and a hierarchical randomised search is used to generate motions. In [5] they add the control of motion annotation which allows the user to include or to exclude a motion along the time line. Figure 2.5 shows an example of motion synthesised by motion annotation. Their similarity metrics based on temporal correspondence of skeletal
2.1. Example-based Synthesis

Figure 2.4: Human motion data is preprocessed as a two-layer graph-like structure [67].

Figure 2.5: An example of synthesised motion by Motion Annotation with hard constraint of a specific frame. A selected “Push” frame at a specific time is forced in motion “Run” [5].

motion and measure skeletal pose similarity. Their path optimisation is a randomised search does not aim for global optimal solution while the approach presented in this thesis is capable to give the global optimal path.

Kovar et al. [64] construct a directed graph on Motion Capture sequences, referred to as a Motion Graphs, where edges correspond to segments of motion and nodes identify connections between them. Motion segments include original motions and generated transitions. They compute distances between pairs of frames to determine if a transition is possible according to a fixed threshold and then construct the transition by aligning the windows with a rigid body transformation followed by blending. The distance is calculated by computing a weighted sum of squared distances between corresponding points in the two point clouds formed over a window of time. Each point cloud is
Figure 2.6: An example of synthesised results by Motion Graphs [64]. The character animation fits the path and transition points is indicated by where the curve changes colour.

the composition of smaller point clouds representing the pose at each time instance in the window. Synthesis is performed by finding an optimal graph walk that satisfies user-defined constraints. Figure 2.6 shows an example of synthesised skeletal motion fitting to a given path using motion graphs [64]. Gleicher et al. [39] enhanced this approach for game applications by allowing a designer to interact with this process. The similarity metric measures surface similarity in terms of a point cloud driven by the skeleton where the temporal correspondence is natural. The optimal graph walk is found by local search methods which cannot guarantee global solution while we provide an efficient global search method with a reasonable computational cost.

These approaches require temporal correspondence to compute frame-to-frame similarity and then identifying transitions. For 3D video, this requires to solve temporal surface correspondence which is still an open problem. In this thesis, we provide a similarity metric that does not require solving correspondence.

Capturing realistic skeleton motion from the real world is not sufficient to achieve a visually realistic character animation. Researchers have investigated techniques to build the character model from observation of the real world. These methods also reduce the requirement for artist expertise to build a character model and therefore lower the cost. Hilton et al. [45] introduced a technique to build recognisable moving 3D models of
individual people. They reconstruct both the shape and appearance of people from low-cost colour images taken in four orthogonal views, a generic 3D human model representing both shape and kinematic joint structure. Therefore, Motion Capture data could be used to drive this model to achieve realistic skeletal motion. Finally, colour texture is projected back to the deformable model to achieve a realistic look. This technique led to the first commercial booth system for capturing animated model of people for applications in games, multimedia and virtual reality [44]. Starck and Hilton [99] extend this to simultaneous multiple view capture arbitrary pose and refinement of reconstruction using stereo.

Ma et al. [77] proposed a method to build animation model of real human body from surface scanned data. The human model is represented by a triangular mesh and described as a layered geometric model. The model consists of two layers: the control skeleton generating body animation from MoCap data, and the simplified surface model providing an efficient representation of the skin surface shape.

Hornung et al. [46] presented a method to animate photos of 2D characters using 3D motion capture data. Given a single image of a person or essentially human-like subject, the motion of a 3D skeleton is transferred onto the subject's 2D shape in the image space, generating a realistic movement. They reconstruct a projective camera model and a 3D model pose which best matches to the given 2D image. A 2D shape template is then fitted onto the character in the input image to drive the deformation according to projected 3D motion data. The technique is demonstrated by a set of human and non-human characters with different motions.

However, Motion Capture driven animation systems even with character models based on real people lack surface dynamics during motion which limits the visual realism they achieve. To better reproduce reality in a virtual world, techniques that capture shape, appearance and surface motion together as a 3D video are investigated and reviewed in the following section.

2.1.3 3D Video

The recovery of dynamic scenes of people's performance from multiple camera observations has received considerable interest in the last decade. 3D video capture and
reconstruction techniques are first reviewed and then the methods to reuse their captured results.

Kanade et al. [60] capture an actor's performance from multiple camera views and reconstruct it in 3D, allowing users to select their own viewpoints at view time independent of the actual camera positions used to capture the event. The scene is captured in a hemi-spherical dome equipped with 51 cameras from different angles. At each time instant, the system constructs a separate 2.5D depth map for each camera view using the multiple-baseline stereo technique introduced by Okutomi and Kanade [90]. Instead of merging these depth maps into a single 3D model, they only merge renderings of the object defined by these depth maps to get a synthetic view of the object. Their approach requires a large number of views to accurately reconstruct a scene and the stereo-alone method may fail in regions without enough texture and borders where the correspondence estimation is unreliable. In addition, constructing a full 3D model of the object is more useful than only a synthetic view of the object since it allows the model to be visualised and manipulated with more ease in both 3D shape and appearance.

Moezzi et al. [83] provide a 17-camera system that performs a volumetric reconstruction and derive a coloured polygonal surface model for rendering. They recover the object shape at each time instance using a volume intersection method. The scene is first quantised into a volumetric grid (voxels). All voxels are iterated through to determine if occupied by the object. Those voxels falling outside of the projected silhouette in any reference view are eliminated from the volume. The ultimate result of this is a shape called the object's visual hull introduced by Laurentini [66]. Finally, the visual hull is converted into a triangle-based surface representation and the texture is then projected back from the real camera images. This approach is capable to create a full 3D model at each time instance. However, a discrete representation of the visual hull will introduce quantisation artifacts in the reconstructed 3D shape of a scene.

Matusik et al. [80] describe a technique to reconstruct the object shape from multiple-view image silhouettes called Image-based Visual Hull (IBVH) and demonstrate the real-time reconstruction of a moving person from four cameras. They perform the volume intersection efficiently in the image space and resulting in a view-dependent
visual hull representation. The synthetic view is then rendered based on the depth to the view-dependent visual hull. This approach removes the quantisation in a voxel representation, hence, eliminate the quantisation artifacts. However, explicitly reconstructing the full 3D model is not provided.

Franco and Boyer [35] provide an efficient algorithm to compute a polyhedral visual hull in the form of a surface mesh from multiple-view silhouettes. They first define the visual hull polyhedral surface as a union of several primitives including viewing edges and view cone intersection edges. They compute all of these primitives, identify and tessellate faces of the polyhedron. The result is a manifold, watertight visual-hull surface enclosing the object. This approach is demonstrated to reconstruct the 3D video of a person dancing from a 19 camera multiple-view video stream. This approach reconstructs a full 3D model from multiple-view silhouettes at each time instance and ensures good topological properties of the surface such as manifoldness which previous methods did not address. However, in general, visual hull alone cannot guaranteed to be the same as the original object since silhouette information alone cannot represent concave surface regions.

Starck and Hilton et al. [102] present a surface motion capture system to capture and reconstruct human performance in 3D combining visual hull and stereo techniques to achieve better quality. Their system has a number of eight high-definition (HD) cameras spaced equally around a circle. At each time step, a visual hull is first derived from multi-view silhouettes and followed by surface refinement. The surface refinement is performed by wide-baseline stereo matching to refine the silhouette images and maximising the photometric consistency between views. The resulting 3D mesh is robust without losing visual detail such as creases in clothing. Texture could be mapped back to the 3D model in a view-dependent way maintaining the highest-resolution appearance. Figure 2.7 illustrates the process of capture, reconstruction and 3D visualisation. This technique has the potential to create a highly realistic 3D video, therefore, results of this work is used throughout the thesis although other sources of 3D video could be used.

These model-free techniques do not require priori knowledge of scene structure and are able to capture arbitrary deforming surfaces, e.g. the complex geometry of clothes and
Chapter 2. Literature Review

Figure 2.7: The process of Surface Motion Capture [102].

hair. However, they do not necessarily have a temporally consistent structure. Separate geometric models constructed at each time-frame are difficult to be instrumented with a skeleton for animation, limiting the techniques to replaying a capture event. To achieve the temporally consistent structure, previous techniques require either a prior model limiting the range of surface deformations, or have to solve the problem of dense surface correspondence for 3D video sequence with free-form deformation. Estimating correspondence has been the subject of much recent work [100, 101, 1, 2, 112, 114, 120].

Starck and Hilton [100] introduce spherical matching to estimate dense temporal correspondence of non-rigid surface shape and appearance of people reconstructed from multiple-view video. The reconstructed body shape is first parametrised in the spherical domain by mapping a uniformly sampled surface onto the unit sphere while preserving the surface sampling rate. Correspondence is then derived in the 2D spherical domain as a bijective mapping, a continuous one-to-one mapping for the entire surface. To solve this bijective mapping between two surfaces, a multiple-resolution coarse-to-fine algorithm is performed by minimising a cost function which takes into account the disparity in surface shape, orientation and colour whilst maintaining a regular sampling. The approach guarantees a continuous one-to-one surface correspondence without overfolding. However, the approach is restricted to surfaces with genus-zero topology, which
2.1. Example-based Synthesis

Figure 2.8: Correspondence labelling for wide-timeframe matching between two frames from different surface motion sequences [101].

limits the application to solve temporal correspondence for 3D video in general. Starck and Hilton [101] address wide-timeframe surface correspondence that does not require a prior model or temporal tracking. A set of local feature descriptors with respect to the corner, edge and region are introduced and constructed for each surface. These descriptors are invariant to isometric deformations allowing large changes in geodesic distance which can occur in practice with changes in surface topology for articulated motions. Correspondence labelling between two surfaces is then performed by finding the most likely assignment between a set of feature points on a source surface and target one. A sparse-to-dense labelling strategy is introduced in which corner features are matched first to achieve surface registration, then edge and region feature points are subsequently matched. Performance is evaluated on 3D video sequence of a moving person with loose clothing. Figure 2.8 shows an example of matching two frames from different surface motion sequences. This approach is capable to handle topology changes and could be incorporated into the animation system presented in this thesis to produce smooth transitions between 3D video sequences.

Ahmed et al. [2] present a dense 3D correspondence finding method for a sequence of individually reconstructed surface meshes of a moving subject. The key is to establish dense surface correspondences between adjacent frames. Sparse 3D correspondences
between subsequent pairs of surfaces are computed by matching 3D positions of optical features. For each frame, they compute SIFT [72] feature locations in all camera views and map them back to the 3D surface. These sparse correspondences are then used as control points to anchor bivariate scalar functions on each reconstructed surface mesh. Dense correspondences are then obtained by matching these function values. By applying this procedure subsequently to all pairs of frames one shape can be aligned with all others and the original input can be reconstructed as a sequence of temporal consistent meshes. However, this approach cannot guarantee a valid one-to-one mapping: local foldovers may occur when triangles are mapped between surfaces. This problem may propagate over time and eventually corrupt the 3D model. Additionally, this approach is not proved to handle changes of the surface topology and requires that topology changes are avoided or corrected during the initial reconstruction step.

Varanasi et al. [112] address the problem of the full tracking of an unknown mesh undergoing deformations with possible topology changes using multiple view videos. They cast the problem as a mesh evolution over time. Both photometric and geometric cues are exploited for robust feature matching. Such an evolution is driven by 3D dense displacement fields estimated between meshes recovered independently at different time frames. SURF [8] descriptors are used as image features and normalised geodesic integral [43] as mesh features. This gives a sparse correspondence between successive frames and defines a sparse displacement field. The dense displacement field is then obtained by propagating the sparse one over vertices by Laplacian diffusion [24] to preserve local shape details. Figure 2.9 shows an example of recovering dense surface correspondence between surfaces where topology changes. Their mesh evolution does not require a known model and is capable of dealing with topological changes.

Zaharescu et al. [120] introduce a 3D feature detector MeshDOG and a 3D feature descriptor MeshHOG for robust surface feature detection and matching as a follow-up to the earlier work [112]. MeshDOG as a generalisation of the Difference of Gaussians (DOG) operator [73] seeks the extrema of the Laplacian of a scale-space representation of any scalar function defined on a discrete manifold based on colour and curvature to consider both photometry and geometry. These detected extrema are thresholded and unstable ones are also eliminated. This gives a set of sparse surface feature points.
2.1. Example-based Synthesis

MehsHOG as a generalisation of the Histogram of Gradients (HOG) descriptor [23] is defined to capture the local geometric and/or photometric properties which is robust to changes in orientation, rotation, translation and noise. An intuitive greedy heuristic is then performed to select a set of best matches between source and target surface meshes. This gives a robust sparse surface correspondence which could be used to derive a dense surface correspondence.

Although these techniques could be incorporated into Surface Motion Graphs to achieve smooth transitions, reliably accurate dense correspondence of dynamic surfaces remains an open problem and beyond the scope of this thesis. The following paragraphs review model-based techniques to reconstruct temporal consistent structure for 3D video.

Starck and Hilton [99] present a framework to capture animated models of people in a multiple camera studio. A prior humanoid model is deformed to match the pose seen in multiple viewpoints at each time frame. The deformation is performed by optimising the geometry and appearance of the model over time with respect to silhouette, stereo and feature cues. They provide the means to capture a dynamic scene with a consistent model that is instrumented with an animation structure to edit the scene dynamics or to synthesise new content. However, this approach requires manually defined feature points in order to pose the model and cannot handle the change of topology.

Carranza et al. [17] describe a model-based system that uses multi-view video footage of an actor’s performance to estimate motion parameters and to interactively re-render
the actor's appearance from an arbitrary viewpoint. The multi-view silhouettes are used to determine the pose for a generic human body model at each time step. The model consists of a triangle surface mesh and a rigged kinematic skeleton. The motion parameters estimation is performed offline by optimising the overlap between the projected model silhouettes and the input image silhouettes. The texture is created by blending the projection of all available camera images onto the geometry. The estimated motion parameters can be further used to animate the model of a different actor or creature. However, one consistent texture from the input video images are used for rendering gives less realistic impression. A high-quality view-dependent texture mapping is preferred, in that case, the accurate object geometry is required which is not possible in the proposed scheme.

Matsuyama et al. [79] propose a framework for dynamic 3D shape reconstruction from multi-view images using a deformable mesh model. They represent the shape by a surface mesh and the motion by translation of its vertices. Their model combines visual hull and multi-view stereo to achieve stability and accuracy of reconstruction. The deformation is performed with respect to multi-view silhouettes and photometric consistency. The result is a topologically consistent surface mesh animation with per-vertex-correspondence. Finally, a view-dependent vertex-based method is used to render object images in arbitrary view. This approach is based on optical flow to solve for dense matches, therefore, the deformable mesh model may fail to follow motions of less textured surfaces such as skin areas. The deformable mesh cannot cope with overall topological structure change such as the hands attached to the body.

Vlasic et al. [114] propose a technique to extract a mesh animation with full correspondence from multi-view video sequences. The system starts with a template mesh rigged with a skeleton, tracks the skeleton across all frames and then deforms the template mesh for each frame. The skeleton tracking is performed by optimising the fit of the skeleton to the visual hull at each time step. The deformation is performed in the respect of matching silhouettes in all views to avoid artifacts at bent joints. They demonstrate the propagation of texture and geometry change on the template mesh throughout the whole motion sequence. This approach captures meshes in full correspondence making them readily usable for 3D manipulation. However, the system may
fail on tracking complex motion and in such cases manual intervention is required to correct the skeleton. This approach may not work for an actor in large loose clothing since both the skeleton tracking and surface deformation will become too ambiguous to solve. In addition, a more complex task of motion editing, e.g. synthesise novel motion, is not investigated.

Aguiar et al. [1] provide a system to capture and reconstruct human performance from multi-view video. They reconstructed spatio-temporally coherent 3D geometry by first laser scanning the performer in 3D as a template and then deforming the template to capture the motion. The model-based deformation capture has a resolution limitation which results in some of the high-frequency detail, such as fine wrinkles in clothing or details of the face, being “baked in” by the laser-scan and not changing according to the actual motion. Although some details may not be true in reality, the performance capture still creates a life-like human motion and 3D video. Figure 2.10 shows a frame of original capture image and reconstructed 3D geometry.

These techniques [79, 102, 1, 114, 35] create 3D video at a quality that approaches the original video images. However, the motion in the form of 3D video can only be replayed and the manipulation or reuse the captured motion is not investigated. Only a few works have been focused on 3D video editing and are reviewed in following.

Starck et al. [103] manually identify transitions to construct a motion graph for a 3D
Figure 2.11: Surface motion synthesis by manually constructing a motion graph [103].

video database. The user can control the motion and viewpoint while viewing synthesised character animation. However, the manual work is time consuming and limits the utility for larger databases. The system only performs transitioning between motions according to user input and does not allow a user to control character location and timing which are critical in animation production. Figure 2.11 shows a manually constructed motion graph and an example of synthesised transitions between walking and idle motion. The framework introduced in this thesis automatically identifies transitions and constructs a motion graph. Path optimisation introduced in this work allows the user to specify high-level control over the production of realistic character animation.

Xu [119] et al. provided a framework for motion editing for 3D videos. They compute shape histograms in a spherical coordinate system to measure frame-to-frame similarity and then use this to identify transitions and construct a directed graph. They allowed users to interactively concatenate frames to desired animation. However, they do not provide any optimisation to satisfy user high-level control which limits scaling to a large database. Figure 2.12 shows examples of transitions for concatenative motion editing. Their approach requires human intervention to create new 3D video and does not optimise the motion to satisfy user-defined constraints. In contrast, the proposed approach is fully automatic to synthesise desired motion with respect the user specified high-level constraints.

For reuse of 3D video of human motion in animation, a natural idea is to be analogous
2.2 3D Shape Similarity

The problem of measuring 3D shape similarity has been widely studied in the content-based 3D object retrieval literature. There are four main classes of techniques: feature-based, graph-based, view-based and bending-invariant methods.

2.2.1 Feature-based Methods

The feature-based approach is general and can be applied to any multimedia database. A feature vector is constructed to describe a particular characteristic or set of characteristics for an object. They may extract global features, local features, or a distribution of features. The similarity between two objects is then defined by a similarity measure between feature vectors. Typically, similarity is computed as a distance measure, where

Figure 2.12: Two examples of transitions for concatenative motion editing [119]. The dot arrow denotes the transition between motions.

to the reuse of Motion Capture sequences. In this thesis, we present a framework that automatically identifies transitions by accurately measuring similarity between 3D video frames, constructs a graph and optimises a path according to user-defined high-level constraints. The presented framework does not require temporal correspondence across the 3D video sequences and can be applied to 3D video captured by both model-free and model-based techniques. This makes the reuse of 3D video more flexible and scalable for animation production. The following sections will review similarity metrics that could be used to identify transitions.
2.2.1.1 Global Feature

Global features are used to characterise the overall shape of the 3D models. Typical global features include: volume, surface area, moments, Fourier coefficients and Wavelet coefficients. Such global features are relatively simple to compute but do not provide discrimination at a local level. In a 3D model matching application such techniques allow fast rejection of incompatible shapes to decrease the range of candidates. Zhang and Chen [122] propose an algorithm to calculate volume, moments and Fourier coefficients of a 3D model directly from a surface mesh representation, by first finding them for the elementary shape (a tetrahedron in the 3D case) then doing a signed sum. This approach is efficient and potential to be used for aligning and even retrieving 3D models. However, the discrimination is not sufficient to match time-varying dynamic surfaces of a human body.

Paquet et al. [92] describes an efficient way to represent the coarse shape, scale and composition properties of an object. They provide three global feature-based descriptors for 3D shape matching, a cord-based descriptor, a moment-based descriptor and a wavelet-based descriptor. These representations are invariant to resolution, translation and rotation. This approach differentiates objects of different classes such as fishes.
and cars, while it does not distinguish different poses of the same person in 3D video sequences. Accurately matching poses of the same person in 3D video sequences is one of the key problems this thesis aims to solve.

Corney \textit{et al.}\cite{22} use three convex-hull based indices hull crumpliness (the ratio of the object's surface area to that of its convex hull), hull packing (the percent of the convex hull volume not occupied by the object) and hull compactness (the ratio of the surface area cubed over the volume of the convex hull squared) for coarsely filtering candidates prior to a more detailed analysis. In addition, a number of other global features including volume, surface area, aspect ratios, crinkliness \cite{86}, compactness, number of facets and number of holes, are also considered. This approach can be integrated into conventional 3D object retrieval system, while may not be useful to 3D video since for a same person these global features are very similar.

Kazhadan \textit{et al.}\cite{61} present a reflective symmetry descriptor that extracts the global symmetry information. The surface mesh for the model is first converted to a 3D density function sampled on a regular voxel grid. Then the reflective symmetry distance of the model for every plane passing through the centre of mass is computed. These distances are combined to obtain the reflective symmetry descriptor for the 3D model. Figure 2.13 illustrates the process to generate this descriptor. Their approach provides a measure of global shape similarity that is orthogonal to above mentioned methods and can be combined with them to provide a better classification performance. However, this approach is still not discriminative enough to distinguish similar poses performed by the same person, for example, a walking pose and a jogging pose.

All of these global shape descriptors are computed efficiently while provide a coarse discrimination. They are useful to classify objects of difficult classes but not suitable to distinguish similar poses of the same person in 3D videos for the purpose of identifying transition points for animation production.

\textbf{2.2.1.2 Local Feature}

Local features can give a more distinctive similarity measure. Previous works are reviewed in following.

Shum \textit{et al.} \cite{97} define similarity as the Euclidean distance between the local curvature
distribution over the mesh representation for two 3D objects. This technique is limited to genus-zero objects with regularly sampled mesh nodes such that the curvature distribution can be correlated. However, curvature distributing over the surface probably remain the same for different poses of the same person. Additionally, there is no guarantee that a 3D video only contains genus-nonzero objects, so this approach cannot be used without losing stability.

Mokhtarian et al. [85] estimated surface Gaussian and mean curvature values at multiple scales together with curvature zero-crossing contours. These features then were utilised for 3D surface matching and object recognition which is an extension of Curvature Scale Space (CSS) for 2D Shapes [84]. This technique tolerates surface noise, small surface changes, independent of the underlying triangulation and applicable to incomplete surfaces which arise during occlusion or to surface with holes. This approach works for classifying objects in classes while does not for distinguish poses of the same person since articulated pose change not necessarily affects the mean curvature on the surface.

Zaharia and Preteux [121] present the 3D Shape Spectrum Descriptor (3D SSD), which is defined as the distribution of a shape index over the entire mesh, to provide an intrinsic shape description of a 3D mesh. The shape index is a local geometric attribute of a 3D surface, expressed as the angular coordinate of a polar representation of the principal curvature vector. This method requires a non-trivial pre-processing phase for meshes that are not topologically consistent or not orientable. The captured 3D video is not necessarily topologically consistent. In contrast, the topology probably changes many times for a complex motion of a person, for example, a dancing.

Chua and Jarvis [20] provide a point signature to describe 3D free-form surfaces that is invariant to rotation and translation. A point signature is computed by accumulating surface information along a 3D curve in the neighbourhood of a point. The point signature can also be used to hypothesise the correspondence between model points. This approach could be used to represent the geometric characteristics of a 3D video frame while a similar technique, Spin Image, which is more discriminative has been used.

Johnson and Hebert [58] present a 3D shape-based object recognition system using
2.2. 3D Shape Similarity

Spin Images. A 3D mesh is described as a set of 3D surface points and normals. Each oriented surface point has an associated spin image. A spin-image is constructed by spinning a plane about the point normal and accumulating the surface points in discrete bins. This process is illustrated in Figure 2.14. By matching spin images, correspondences between 3D surface points can be established and used to match 3D models independent of the transformation. Two 3D models are said to be similar when many 3D surface points are similar. However, this requires many spin images implying considerable storage requirements, and preventing their use directly as descriptors for retrieval purpose. Original Spin Image requires computing a signature image for all surface points which is of prohibited computational complexity for high-resolution 3D video frame to accurately match 3D video frames with the same shape.

Assfalg et al. [7] present a 3D object retrieval approach based on spin images by constructing a set of feature vectors to characterise the spin images for a model. They define three independent sets of regions for the spin images: sectors of circular crowns for both the half-plane and circular sectors as shown in Figure 2.15. Each spin image is re-binned as a multi-dimensional feature vector. Feature vectors for a model are then extracted by a clustering process. They provide a less computational Spin Image but does not necessarily a better discriminative performance.

2.2.1.3 Local Feature Distribution

Local features are compared using a descriptor of the feature distribution. The distribution of local features is preferred since it is more robust to noise and more discriminative. Osada et al. [91] introduced a Shape Distribution as a signature to discriminate similar and dissimilar models. The Shape Distribution is a probability distribution sampled
Figure 2.15: The compound object descriptor comprises descriptors[7]. For (a) $np$ crowns with $\beta > 0$; (b) $nn$ crowns with $\beta < 0$; (c) $ns$ sectors.

from a shape function measuring global geometric properties of an object. A Similarity Measure is computed as the difference between Shape Distributions, which is invariant to rotation and tessellation of the 3D polygonal model. However, the shape distribution can only reveal gross shape similarity of objects. Figure 2.16 illustrates the process to generate Shape Distribution descriptor. This descriptor can be efficiently computed, handle with topology changes, robust to noise and subtle surface changes. Therefore, this approach is considered and evaluated in this thesis.

Ankerst et al. [3] use a 3D Shape Histogram as a shape signature to classify a molecular database. A 3D Shape Histogram is based on a partitioning of the space where an object resides, that is, a complete and disjoint decomposition into cells which correspond to the bins of the histogram. There are three typical decompositions: a shell model, a sector model and a spiderweb model. Shape Similarity is computed as the difference between histograms. Figure 2.17 shows an example of Shape Histogram with different configuration. This approach is robust to topology changes, surface noise and surface subtle changes. However, the descriptor is not invariant to rotation, so a rotation-invariant comparison scheme is required. This approach is also considered and evaluated in this thesis.

Belongie et al. [9] introduced Shape Context for 2D/3D matching. The Shape Context descriptor provides a shape histogram centred at a surface point. The cost of match-
2.2. 3D Shape Similarity

Figure 2.16: The generation of Shape Distribution[91].

ing two sample points from two models is defined as the distance between their 3D shape contexts. Correspondence is found by minimising the total cost in matching all sample points from two 3D models. Surface correspondence can be used to first align two models and then compute the distance between two surfaces. However, surface correspondence is inherently ambiguous and point matching is an expensive process. Figure 2.18 shows an example of computing and matching Shape Context descriptors.

Ohbuchi et al. [89] introduce two further shape descriptors, Angle Distance (AD) and Absolute Angle Distance (AAD) histograms for 3D matching. Both AD and AAD take into account not only the distance between the point pair but the angle formed by the surface normal vectors at the pair of points. The surface normals of 3D video are not necessarily exactly correct and therefore this approach may not be stable across the whole sequences.

All of these descriptors based on distribution of local features provide a more robust and more discriminative means of measure similarity than using local features only. Some of these techniques [91, 3] are considered and evaluated in this thesis with some modification to better adapt 3D video data.
Figure 2.17: The generation of Shape Histograms [3]. From top to bottom, the number of shells decreases and the number of sectors increases.

2.2.1.4 Compact Canonical Representations

Another popular approach is a transform-based representation which describes shapes in a transformation invariant manner. Kazhdan et al.[62] propose Spherical Harmonic Descriptors that are invariant to rotation for 3D shape retrieval. A 3D model is regarded as a spherical function which can be decomposed to the sum of its harmonics. Summing the harmonics within each frequency to get each frequency energy component, the norm of which is the amount of energy the 3D model contains at each frequency. Spherical Harmonic Descriptors are invariant to rotation, because applying a rotation to a spherical function does not change its energy representation. Figure 2.19 illustrates the process to generate a Spherical Harmonic Shape Descriptor. However, the representation has a potential ambiguity problem as illustrated in Figure 2.20. The frequency decomposition is performed independently in concentric spheres, such that two different shapes can have the same spherical harmonic representation.

Novotni and Klein [88] provide 3D Zernike Descriptors for 3D shape retrieval, which is an extension of the Spherical Harmonic Representation [62]. A set of descriptors are
Figure 2.18: The computation and matching of Shape Context [9]. (a) Shape 1; (b) Shape 2; (c) Histogram bins used to compute Shape Context, 5 shells and 12 sectors; (d) Shape Context for the sample marked by a sphere; (e) Shape Context for the sample marked by a diamond; (f) Shape Context for the sample marked by a triangle. (g) Correspondences found using bipartite matching. Note the visual similarity of the Shape Contexts for a sphere and a diamond.

obtained that are orthonormal, complete and rotation invariant. However, 3D Zernike Descriptors suffer the same ambiguity problem with Spherical Harmonic Descriptors [62].

2.2.2 Graph-based Methods

Graph based techniques provide a shape description based on the surface topology of an object. These techniques often require complex extraction algorithms and matching strategies to define the similarity between graphs.

2.2.2.1 B-rep Graph

In the CAD industry, the most common used graph-based representations are Boundary Representation (B-rep) and Constructive Solid Geometry (CSG). A B-rep describes a
3D model in terms of its faces, loops, edges and vertices. An example of B-Rep Graph descriptor is shown in Figure 2.21. CSG describes a 3D model in terms of a set of Boolean operations applied to primitive geometric entities, such as cubes and cylinders. However, very few commercial CAD systems support CSG.

El-Mehalawi and Miller [30] construct an attributed graph from a B-rep and measure similarity by using an inexact graph matching algorithm. The nodes of the graph represent model surfaces and the edges of the graph represent model edges. Therefore, the graph itself represents the topology and the attributes represent the geometry of the model. However, this detailed representation makes it impractical for large and complex models.
2.2. 3D Shape Similarity

Figure 2.20: The ambiguity of Spherical Harmonics Representation. The models differ by more than a single rotation, their rotation invariant representations are the same [62].

Figure 2.21: An example of B-Rep Graph descriptor [30]. From an example CAD part to its B-Rep data structure.

Similarly, McWherter et al. [81] compare models based on shape using information extracted from B-rep into Model Signature Graphs (MSG). A MSG is essentially an attributed graph with the vertices representing the faces of the model and the vertex attributes describing the qualities of the face, such as the type of surface (flat, curved), relative size, the surface area and a set of surface normals. Figure 2.22 shows an example of the model and corresponding MSG descriptor. Spectral graph theory is used as a basis to approximate graph similarity among model signature graphs. The eigenvalues of the adjacency matrix of the graph are sorted to produce a graph spectrum which is strongly related to the topology of the model. A similarity measure is computed as the Euclidean distance between the graph spectrum. However, these methods are usually limited to the CAD community, for example, in matching mechanical parts.

Zuckerberger et al. [125] decompose a polyhedral surface (mesh) into “meaningful”
patches and use similar methods to construct a graph representation of the model for content-based retrieval and model simplification. These techniques focus on retrieving 3D models of industrial products, where the object is rigid and the topology does not change. It is difficult to directly adopt or extend them to measure similarity of a non-rigid time-varying surface of the human body where the topology may change over time.

2.2.2.2 Skeletal Graph

Sundar et al.[104] use a skeletal graph which encodes both the geometric and topological information in the surface to match and compare 3D models. Surface models are first converted to a volumetric representation. Then volumetric thinning is used to obtain a set of skeletal points which are connected in an acyclic shape graph using the Minimum Spanning Tree algorithm. Each node in the graph represents a segment of the original skeleton, geometrical signature vector and a topological signature vector are encoded into the node. This topological signature vector is defined recursively over the subgraphs of the node using eigenvalues of their adjacency matrices. Skeletal graphs
are compared in a hierarchical manner using a recursive depth-first search algorithm by computing topological and local shape similarity. Matching two graphs is formulated as a largest isomorphic subgraph problem and then reformulated to find a maximum cardinality, minimum weight matching in a bipartite graph. Evaluation of the skeleton is proportional to the number of voxels in the model of spanning tree which gives efficient computation. Skeleton matching can do part matching and articulated matching. Furthermore, since it outputs node-to-node correspondences, this information can be used to align skeletons and models. An example of skeleton matching is showed in Figure 2.23.

Iyer et al.[56] and Lou et al.[71] use similar methods to retrieve 3D engineering models. A 3D model is converted into a voxel model and then into a thinned skeleton and finally a skeletal graph. Their thinning algorithm iteratively erodes voxels until a one-voxel width skeleton is left. The skeletal graph consists of nodes, edges, and loops. Each node corresponds to the voxel situated at the ends of the edges, each edge corresponds to an independent geometric entity, each loop corresponds to a hole in 3D model, thereby, giving a shape to the model in physical space. Thus, the geometry of the 3D model is captured in the individual entities of the skeleton and the topology in the skeletal graph. In addition, they extract the feature of a 3D model including moments/invariants, geometry parameters, vocalisation parameters and graph parameters. Similarity measure computed as the Euclidean distance of the feature vectors and the distance of skeletal graphs between the 3D models. An R-tree based multidi-
Chapter 2. Literature Review

Figure 2.24: The generation of Skeletal Graph [56]: from a CAD model to Skeleton and then Skeletal Graph[56].

A dimensional index is used to speed up the feature-vector based search operation, while a decision tree-based approach is used for efficiently indexing/searching skeletal graphs. Figure 2.24 illustrates the process of generating the Skeletal Graph descriptor from a 3D model.

Menier et al.[82] propose a method to find the skeleton pose which can best fit the body pose in their motion capture work. Silhouettes are extracted from calibrated cameras with different viewpoints by standard background subtraction techniques. From these silhouettes, the visual hull is computed and discrete medial axes derived to form a 3D skeleton. An expectation maximisation(EM) approach is used to fit an a-priori skeletal articulated model to the medical axes data. This approach is very efficient and robust to noisy silhouettes. Figure 2.25 illustrates the process of recovering skeleton pose from multiple-view video input. However, a skeletal articulated model whose dimensions are manually set is required, and their skeleton sequences lack smooth temporal continuity.

These techniques require either fitting a skeleton onto the 3D model or extracting a skeleton from the 3D model, which is difficult for 3D video where the topology of the surface of human body may dramatically change over time. Reliable temporally consistent fitting or extraction of a skeleton for a sequence of 3D video still remains an open research problem. In addition, it is difficult for skeleton-based methods to deal with surface dynamics such as loose clothing. Therefore, in this thesis we focus on non-skeleton methods to avoid solving this problem.
2.2. 3D Shape Similarity

Figure 2.25: 3D skeleton recovery from body pose [82]. From left to right: Original image in a single view; Silhouette in a single view; Visual hull reconstructed from multiple views; Medial axis points, i.e. noisy skeleton; Skeleton pose.

2.2.2.3 Reeb Graph

Hilaga et al.[43] propose a method based on Multi-resolutional Reeb Graphs (MRGs) to estimate a measure of similarity and correspondence between 3D shapes. MRGs can be easily constructed, a MRG using a height function is showed in Figure 2.26. Instead of choosing the height function, Hilaga et al.[43] choose the normalised integral of geodesic distance as the continuous scalar function for rotation invariance and resistance against noise. They resample and subdivide a mesh to make it fine enough to approximate geodesic distances by Dijkstra’s algorithm to construct the MRG directly from a triangle mesh. The similarity is calculated with a coarse-to-fine strategy using the attributes of nodes in the MRG and topological consistency.

Bespalov et al.[11] implement Hilaga’s techniques for CAD model matching. Their experiments demonstrate that the Reeb Graph tends to become less sensitive to topology and at the same time more dependent on geometry for complex structured models, which may affect both efficiency and accuracy of comparisons. Chen and Ouhyoung[18] extend Hilaga’s work to a 3D model retrieval system. In practise, they merge vertices and parts in a pre-processing stage to speed up the search.

Tung and Schmitt[111] provide a flexible multi-resolutional and multicriteria representation called the augmented Multiresolution Reeb Graph (aMRG) to improve the original MRG method. They introduce a topological consistency criteria and geometric at-
Chapter 2. Literature Review

Figure 2.26: The generation of Multi-resolutional Reeb Graph (MRG) using a height function [43]. In coarse resolution, the object is considered as a node and as the resolution increasing (r is repeatedly divided by 2) the connected components (S) forms new nodes (n) hierarchically.

Figure 2.27: Improvements of Augmented Multiresolution Reeb Graph (aMRG) [110].
tributes to the nodes in order to obtain better matching between nodes of graphs in comparing models. For example, their method avoids false matching between legs and arms in a human model as shown in Figure 2.27. They first propose a node similarity function which considers 2D/3D object features (relative volume, cord length, curvature and colour) then a graph similarity function giving an improved estimation of the similarity between models. Experiments on eight different classes of objects show that the aMRG method outperforms six other tested methods, including a cord histogram, D2 shape distribution [91], 3D shape spectrum descriptor, 3D Hough Descriptor, Complex EGI, and the surface/volume ratio methods.

These techniques could be incorporated in this thesis to provide a similarity metric. However, they have only been tested for classifying objects into different classes and not for accurately matching time-varying surfaces of a human motion for concatenating 3D video as novel animation, which is a more challenging task. The evaluation of these techniques by measuring similarity for 3D video sequences is interesting and will be left for future work.

2.2.3 View-based Methods

View-based methods represent objects by their image-plane projection. The basic idea is that if two 3D objects are similar, their image projecting will be similar too. Solving the 3D shape similarity becomes iteratively solving the 2D image similarity.

Heczko et al. [41] present a silhouette descriptor to characterise 3D objects in terms of their silhouettes obtained by parallel projections. This method requires a PCA-normalised preprocessing: the first and the second principal axis are computed and then used to define the coordinate system; the size of a 3D model is normalised by dividing the radius of the bounding sphere. This gives a rotation and scale invariant 2D descriptor. A view is represented as a feature vector of the Fourier coefficients for a regularly sampled set of points on the silhouette contour. Heczko et al. [41] propose a similar descriptor called the depth buffer descriptor in which similarity is computed as the distance between the depth buffer feature vectors of two 3D models.

Chen et al. [19] introduce the LightField Descriptor in which the appearance of an object is characterised by the projected appearance in a set of camera views. Similarity is
Figure 2.28: An example of comparing Light Field Descriptors [19]. (a) Two 3D models. 20 images are rendered from vertices of a dodecahedron. All the corresponding 2D images from the same viewing angles are compared, a similarity value under this rotation of camera system is obtained. Repeat this process (b), (c), (d), a rotation of camera system with the best similarity (highest cross-correlation) is found and the similarity is the summation of the similarities among all the corresponding images.
2.2. 3D Shape Similarity

computed by rotating the camera system surrounding each model until the highest overall similarity (cross-correlation) between the two models from all sampled viewing angles is reached. The similarity between two 3D models is defined as the summation of the similarities across all the corresponding 2D images. Figure 2.28 shows an example of matching two 3D models using LightField descriptors. This descriptor and its extension to colour domain have been investigated and evaluated in this thesis. The detailed comparison with proposed Shape-Colour Histograms is presented in Chapter 5.

In general, view-based methods are sensitive to self-occlusion and to achieve a discriminative similarity measure usually requires generation and storage of a large number of 2D image projections, which is computational complex and storage inefficient.

2.2.4 Bending-invariant Methods

Bending-invariant techniques have been proposed to retrieve similar objects independent of changes in articulated pose.

Elad and Kimmel [31] present a method to construct a bending invariant signature for these models. They utilise the geodesic distance between surface points as an invariant to surface bending. A bending invariant surface is generated by transforming the geodesic distances between points into Euclidean ones (via an MDS procedure). They translate the problem of matching non-rigid objects in various postures into a simpler problem of matching rigid objects. Figure 2.29 shows an example of transforming a human body in different articulated pose into a same signature.

Jain and Zhang [57] present an approach to robust shape retrieval from databases containing articulated 3D models. Each shape is represented by the eigen vectors of a shape affinity matrix defining the geodesic surface distance between model points. This gives a spectral embedding which achieves normalisation against rigid-body transformations, uniform scaling, and shape articulation.

These techniques are useful when the application does not differentiate the same object with different articulated poses. However, in contrast, the application of concatenating 3D video as novel character animations requires differentiating different poses of the same person. Therefore, these techniques are not suitable and not investigated in this thesis.
2.3 Shape-Colour Similarity

Although both shape and colour information are important for 3D object recognition, retrieval and matching, comparatively little research has been done on combining both of them.

Higuchi et al. [42] proposed a method for representing both colour and shape information using a common framework, the Spherical Attribute Image (SAI), to map the curvature and colour at each vertex of a surface mesh onto a spherical image for 3D object recognition. By finding the rotation that brings spherical images into correspondence a model object and an observed surface can be matched. Their method took into account the actual photometric and geometric information-distribution on the surface and showed improved recognition performance.

Matas et al. [78] proposed Colour Adjacency Graph (CAG) as an object representation to recognise objects with multiple colours. Each node of the CAG represents a single chromatic component of the image defined as a set of pixels forming a unimodal cluster in the chromatic scattergram. Edges encode information about adjacency of colour components and their reflectance ratio. The CAG is related to both the histogram and region adjacency graph representations. The performance of the approach was demonstrated on a range of difficult object recognition and localisation problems involving
2.3. *Shape-Colour Similarity*

Shape-Colour Similarity

complex imagery of non-rigid 3D objects under varied viewing conditions.

Slater and Healey [98] derived invariants of local colour distributions that are independent to views and illumination associating geometric information for hypothesis verification and pose estimation in an object recognition. These invariants capture information about the distribution of spectral reflectance which is intrinsic to a surface and thereby provide substantial discriminatory power for identifying a wide range of surfaces. These invariants can be computed efficiently from colour image regions without requiring any form of segmentation. They demonstrated the system's ability to recognise model objects in cluttered scenes.

These techniques have only considered rigid object and focus on classifying to different classes. They are not sufficient to provide the required discriminative to measure shape-colour similarity for a highly dynamic 3D video sequences. In this thesis, we extend Shape Histogram to incorporate colour information and provide a Shape-Colour Histogram as a novel shape-colour descriptor, which is discriminative and aims to matching 3D video sequences.

Chen et al. [19] proposed LightField descriptors to measure visual similarity between 3D models and the main idea is that if two 3D models are similar, they also look similar from all viewing angles. For computational efficiency they only use silhouettes in each view which considers geometric similarity and ignore photometric similarity. We extend LightField to exploit photometric information by using colour views and compare it with the proposed Shape-Colour Histograms in Chapter 5.

The alternative to exploit shape and colour information is combining different representations by kernels. Caputo and Dorko [16] presented a kernel method to combine colour and shape information for object recognition and showed an increased recognition rate. It does not require a new common representation, but uses the power of kernels to combine different representations together in an effective manner. Experiments show improved recognition rate.

Kernel-based methods for object retrieval and recognition has received extensive research. These techniques could be incorporated to provide an alternative to a shape-colour descriptor. However, this is beyond the scope of this thesis and has to be left as future work.
2.4 Temporal Similarity

Temporal shape matching for 3D video sequences has received limited investigation. Related work can be found in the literature on human motion recognition and example-based animation from video [96] or marker-based human motion capture [67, 38, 5], which have been presented in Section 2.1.2. Previous work in human motion recognition are reviewed as follows.

Bobick et al. [12] introduce a temporal template as a static vector-image to construct a view-specific representation of motion over time. This temporal template has two components, Motion-Energy Images (MEI) and Motion-History Images (MHI). The binary-valued MEI represents where motion has occurred in an image sequence and the MHI is a scalar-valued image where intensity is a function of recency of motion. For each view of each movement, a statistical model of the moments (mean and covariance matrix) is generated for both MEI and MHI. Matching is performed by calculating a Mahalanobis distance between the moment description of the input and each of the known movements. This approach is based on video or 2D images sequences, hence, sensitive to background subtraction, viewpoint change and self-occlusion. The discrimination is sufficient to allow recognition of different human motion, but may not be sufficient to accurately identify transition frames for seamlessly concatenating 3D video sequences.

Efros et al. [29] introduce a pixel-wise optical-flow motion descriptor to recognise human actions which are observed in low-resolution image sequence. They measure the optical flow in a figure-centric spatio-temporal volume for each person. They first compute optical flow at each frame using the Lucas-Kanade algorithm [75]. The computed optical flow is then split into two scalar fields corresponding to the horizontal and vertical components, each of which is then half-wave rectified into four non-negative channels and each channel is blurred with a Gaussian. Comparison is then performed by computing the sum of similarity in each motion channel over a period of time. The result is a time-filtered frame-to-frame similarity matrix. Finally, they exploit computed similarity matrix for classification of different actions. The key challenge addressed is to make use of noisy optical flow measurements, which is quite different with ours. Time-filter is used to enforce temporal similarity. Similarly, we also exploit a weighted
2.4. Temporal Similarity

Figure 2.30: Space-time shapes of jumping-jack, walk, and run actions.

time-filter to enforce temporal similarity and provide an adaptive scheme to determine the temporal window size.

Gorelick et al. [40] address the problem of recognising human action in single-view video sequences. They regard human actions as 3D shapes induced by the silhouettes in a space-time volume, extracting space-time features such as local space-time salience, action dynamics, shape structure and orientation. These features are then used for action recognition and classification. Similarity measurement is performed by computing Euclidean distance between these features over a pre-defined time window. Figure 2.30 shows some examples of space-time shapes for different actions. This approach enforces global alignment of individual frames along space and time which improves the performance of action recognition. Similarly, we introduce a novel global-alignment Shape-flow descriptor (SHvrG) to compare two 3D video frames in both shape and motion. Improved discrimination is demonstrated in Section 4.2.2, Chapter 4.

Weinland et al. [115] address the problem of recognising human action in multi-view video sequences. They proposed a free-viewpoint representation based on Fourier analysis of Motion History Volumes (MHV) in cylindrical coordinates. They first separately compute visual hulls from multi-view images and accumulate them over a temporal window into MHVs. These MHVs are transformed into cylindrical coordinates around their vertical axes and view-invariant features are then extracted in Fourier space. Similarity is measured by calculating the Mahalanobis distance between feature vectors. This approach incorporate temporal information to achieve better recognition for human motion in multi-view video sequences, which motivates us to extend 3D shape
descriptors to time domain, although, for a different purpose, we aim to identify 3D video frames as smooth transition for seamlessly concatenating two 3D video sequences.

2.5 Self-Similarity Matrix

In data analysis, the Self-Similarity Matrix is a graphical representation of similar sequences in a data series. Similarity can be explained by different measures, like spatial distance (distance matrix), correlation, or comparison of geometric and/or photometric properties. Foote exploited Self-Similarity Matrix to visualize the time structure of music and audio which allows identification of structural and rhythmic characteristics [34]. Junejo et al. [59] explored Self-Similarity Matrices of action sequences over time for cross-view action recognition and observed the high stability of such measures. A Self-Similarity Matrix can be the starting point for Recurrence Plots [28]. With Self-Similarity Matrix, one can graphically detect hidden patterns and structural changes in data and see similarities in patterns across the time series under study. In this thesis, Self-Similarity Matrix of human motion in 3D video sequences has been used to evaluate the recognition performance of different descriptors by measuring shape, shape-motion and shape-colour similarity.

2.6 Summary

In this chapter, we have reviewed related research to produce realistic character animation from the real world capture using 2D video, skeleton Motion Capture and 3D video. An important step in reuse is the evaluation of similarity between different frames in a sequence. A review of similarity metrics previously used for static (rigid) object retrieval is therefore presented together with metrics used for 3D video surface sequences. Static shape similarity, shape-colour similarity and their combination with temporal similarity are reviewed. In the following chapter, we quantitatively evaluate four state-of-the-art shape-only similarity metrics for the problem of identifying similar shape and motion in 3D video. Similar frames provide possible transitions points to concatenate segments of 3D video while preserving the natural motion.
One of the key advances required to develop an example-based animation system is identifying transitions where different sequences can be seamlessly concatenated. To achieve this, the similarity of frames must be maximised at transitions. The similarity metrics used for Skeletal Motion Capture (MoCap) cannot be directly applied to 3D video reconstructed by Surface Motion Capture (SurfCap) [102]. Measuring human surface similarity requires a shape descriptor that is sufficiently distinct to differentiate articulated poses and non-rigid cloth deformation while tolerant to changes in surface topology for similar poses. 3D shape similarity is a widely known and investigated problem in 3D object-based retrieval literature [106, 14, 36, 55]. Several 3D model search engines have become available [18, 37, 54]. However, these techniques are designed to distinguish objects from different classes, for example, to differentiate a chair from a tank. Query for similar shapes from a single temporal-varying object, in this case, the same actor performing different motions, has received little attention. In the following sections, we compare four state-of-the-art techniques from the shape retrieval literature for the problem of human surface shape similarity; perform a quantitative ground-truth evaluation using synthetic data and a qualitative evaluation for real 3D video data.
Chapter 3. 3D Shape Similarity

3.1 3D Shape Descriptors

We consider shape descriptors based on local-feature distribution to measure similarity for 3D video. These descriptors provide a global object description that allows for changes in captured surface topology. Their performance is evaluated and compared using the Receiver-Operator Characteristic (ROC) in classifying correct versus incorrect similarity against synthetic ground-truth. Four state-of-the-art 3D shape descriptors including Shape Distribution [91], Spin Image [58], Shape Histogram [3] and Spherical Harmonic [62] are evaluated. A brief description for each technique is presented in this section.

3.1.1 Shape Distribution [91]

Shape Distribution (SD) provides a shape signature as a probability distribution of a shape function that measures some geometric properties of a 3D model. Typical shape functions are the angle, distance and area for randomised points on the model surface. Here we adopt the $D2$ measure, the Euclidean distance between two random points on the surface, as proposed by [91]. Similarity is measured as the Euclidean distance between the distribution $D2$ defined for two meshes. Figure 3.1 illustrates the Shape Distribution representation computed for a single frame of a 3D video sequence of a person. Given a 3D triangle mesh representation $M = < V, F >$, where $V$ denotes a set of vertices and $F$ a set of triangles, the Shape Distribution descriptor $SD(M)$ is constructed as follows:

1. Distance is measured between $N_p$ random pairs of points on the surface of a 3D model. Instead of choosing sample vertices of the 3D model directly, we choose re-sampled points on the surface in order to avoid being biased and sensitive to changes in tessellation. For a triangle $f = < v_0, v_1, v_2 > \in F$, a random point $x = (x, y, z)$ is re-sampled as follows,

$$ x = (1 - r_0 - r_1)v_0 + r_0v_1 + r_1v_2 $$

where $v_0, v_1, v_2$ denote vertex 3D position and $r_0, r_1 \in [0, 1]$ are two random real numbers. This gives a set of re-sampled points $X = \{x_0, ..., x_{N_F-1}\}$. $N_F$ is the
total number of triangles in $F$. Euclidean distance $d$ for a pair of surface points $x_i, x_j \in X$ is then calculated,

$$d = |x_i - x_j| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$  \hspace{1cm} (3.2)

where $i, j \in [0, N_F - 1]$ are two random integer numbers. We repeat randomly choosing $i, j$ and computing the distance $d$ for $N_p$ times. This gives a set of sampled distance $D = \{d_0, ..., d_{N_p-1}\}$.

2. A 1D histogram $H(D) = [H_l(D)]_{N_d}$ is created by constructing $N_d$ fixed sized bins, then to count the number of point-pairs at different distances falling into each of these bins.

$$H_l(D) = \sum_{k=0}^{N-1} g(l, k)$$  \hspace{1cm} (3.3)

$$g(l, k) = \begin{cases} 
1 & \text{if } d_k \text{ in } l^{th} \text{ bin} \\
0 & \text{otherwise} 
\end{cases}$$  \hspace{1cm} (3.4)

The $l^{th}$ bin is defined as a subspace,

$$l^{th} \text{ bin} := [l \cdot \Delta d, l \cdot \Delta d + \Delta d)$$  \hspace{1cm} (3.5)

where $\Delta d$ denotes the bin size for the sampled distance.

3. The final descriptor $SD(M)$ is a 1D histogram of the probability normalised by dividing by the total number of samples $N_p$.

$$SD(M) = \frac{H(D)}{N_p}$$  \hspace{1cm} (3.6)

### 3.1.2 Spin Image [58]

A Spin Image (SI) is a 2D histogram which encodes the density of mesh vertices projected onto an object-centred space. Given a 3D surface mesh $M = <V, F>$, the Spin Image descriptor $SI(M)$ is constructed as follows:

1. An object-centred coordinate system is defined by the centroid $c$ and the vertical axis $a$ of the object. For each vertex $v \in V$, the distance $\alpha$ along and the distance $\beta$ from the vertical axis is computed,

$$\alpha = (v - c) \cdot a$$  \hspace{1cm} (3.7)
where function $| \cdot |$ computes Euclidean distance between two 3D points. This gives a set of 2D points $A = \{(\alpha_n, \beta_n)\}, n = 0, ..., N_V$. $N_V$ is the total number of vertices in $V$.

2. A 2D histogram $H(A) = [H_{lm}(A)]_{N_\alpha \times N_\beta}$ is created and dividing the space to $N_\alpha \times N_\beta$ bins, then to count the number of vertices falling into each of these bins.

$$H_{lm}(A) = \sum_{k=0}^{N_V-1} g(l, m, k) \tag{3.9}$$

$$g(l, m, k) = \begin{cases} 1 & \text{if } (\alpha_k, \beta_k) \text{ in } (l, m)^{th} \text{ bin} \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

The $(l, m)^{th}$ bin is defined as a subspace,

$$(l, m)^{th} \text{ bin} := [l \cdot \Delta\alpha, l \cdot \Delta\alpha + \Delta\alpha) \times [m \cdot \Delta\beta, m \cdot \Delta\beta + \Delta\beta) \tag{3.11}$$

where $\Delta\alpha, \Delta\beta$ denote the bin size for $\alpha$ and $\beta$ respectively.

$$\Delta\{\alpha, \beta\} = \frac{\{\alpha, \beta\}_{up} - \{\alpha, \beta\}_{low}}{N_{\{\alpha, \beta\}}} \tag{3.12}$$

Figure 3.1: Illustration of Shape Distribution.

$$\beta = |(v - c) \times a| \tag{3.8}$$
3.1. 3D Shape Descriptors

3. The final descriptor $SI(M)$ is a 2D histogram of the probability normalised by dividing by the total number of vertices $N_v$.

$$SI(M) = \frac{H(A)}{N_v}$$

(3.13)

Figure 3.2 illustrates the Spin Image for a single frame of a 3D video sequence, showing the histogram distribution resulting from a plane rotated about the vertical axis through the centroid of a 3D model.

3.1.3 Shape Histogram [3]

A Shape Histogram (SH) partitions the space containing an object into disjoint cells corresponding to the bins of a histogram. Given a 3D triangle mesh $M = < V, F >$, a volume sampling spherical histogram is constructed as follows:

1. A volumetric representation is constructed by first dividing the space into a $N_g \times N_g \times N_g$ voxel grid and then identify those voxels which lie inside the 3D model. Given a bounding box that is determined by a triple set of length, width and height $(l, w, h)$, the voxel size $\Delta v$ is set as

$$\Delta v = \frac{\max\{l, w, h\}}{N_g}$$

(3.14)
where \( N_3 \) is the number of voxels. We only consider those voxels occupied by the 3D model. If we denote \( o = (x, y, z) \) for 3D position of an occupied voxel centroid. For all occupied voxels, this gives a set \( O = \{ o_n \}, n = 0, ..., N_o - 1 \). \( N_o \) denotes the total number of occupied voxels, \( N_o < N_3 \).

2. Space in Cartesian coordinate system is transformed to a Spherical coordinate system defined by the centre of mass for the model and vertical axis. For each occupied voxel centroid \( o \in O \), the spherical coordinates \( s = (r, \theta, \phi) \) are calculated as follows,

\[
\begin{align*}
    r &= \sqrt{x^2 + y^2 + z^2} \quad (3.15) \\
    \theta &= \arccos \frac{z}{r} \quad (3.16) \\
    \phi &= \arctan \frac{y}{x} \quad (3.17)
\end{align*}
\]

This gives a set of spherical coordinates of occupied voxel centroids \( S = \{ s_k \}, k = 0, ..., N_o \).

3. A 3D spherical histogram \( H(S) = [H_i(S)]_{N_1} \) is constructed, where \( \mathbf{l} = [r, \theta, \phi], N_1 = N_r \times N_\theta \times N_\phi \), accumulating the voxels in the volume representation,

\[
H_i(S) = \sum_{k=0}^{N_o-1} g(l, k) \quad (3.18)
\]

\[
g(lk) = \begin{cases} 
1 & \text{if } (r_k, \theta_k, \phi_k) \text{ in } l^{th} \text{ bin} \\
0 & \text{otherwise}
\end{cases} \quad (3.19)
\]

The \( l^{th} \) bin is defined as a subspace,

\[
l^{th} \text{ bin} := [r \cdot \Delta r, r \cdot \Delta r + \Delta r] \times [\theta \cdot \Delta \theta, \theta \cdot \Delta \theta + \Delta \theta] \times [\phi \cdot \Delta \phi, \phi \cdot \Delta \phi + \Delta \phi] \quad (3.20)
\]

where \( \Delta r, \Delta \theta, \Delta \phi \) denote the bin size for radius, inclination angle and azimuth angle respectively,

\[
\Delta \{r, \theta, \phi\} = \frac{\{r, \theta, \phi\}_up - \{r, \theta, \phi\}_low}{N_{\{r, \theta, \phi\}}} \quad (3.21)
\]

4. The final descriptor \( SH(M) \) is a 3D histogram of the probability normalised by dividing by the total number of occupied voxels \( N_o \),

\[
SH(M) = \frac{H(S)}{N_o} \quad (3.22)
\]
3.1.3D Shape Descriptors

The shape histogram representation is illustrated in Figure 3.3, 3D histogram is converted to 1D histogram for visualisation.

3.1.4 Spherical Harmonics [62]

The Spherical Harmonic Representation (SHR) describes an object by a set of spherical basis functions. A descriptor is constructed by measuring the energy contained in different frequency bands, where the frequency components are rotation invariant. Given a 3D mesh $M$, the Spherical Harmonic Representation $SHR(M)$ is constructed as follows:

1. The surface of an object is first rasterized into a $2N_R \times 2N_R \times 2N_R$ voxel grid, assigning a voxel value of 1 if it is within one voxel width of a polygonal surface,
and assigning it a value of 0 otherwise. To normalise for translation and scale, we move the model so that the centre of mass (average of 3D positions across all voxel centre) lies at the point \((N_R, N_R, N_R)\) and we scale it so that the average distance from non-zero voxels to the centre of mass is \(\frac{N_r^2}{2}\). We use this approach rather than a simpler one based on the centre and radius of the bounding sphere because it is less sensitive to outliers.

2. The voxel grid is regarded as a (binary) real-valued function defined on the set of points with length less than or equal to \(N_R\) and express the function in spherical coordinates:

\[
f(r, \theta, \phi) = \begin{cases} 
1 & \text{if } (r, \theta, \phi) \text{ in an occupied voxel} \\
0 & \text{otherwise}
\end{cases} \tag{3.23}
\]

where \(r \in [0, N_R], \theta \in [0, \pi], \phi \in [0, 2\pi]\). By restricting to the different radii we obtain a collection of spherical functions \(\mathcal{F} = \{f_0, f_1, \ldots, f_{N_R-1}\}\) with

\[
f_r(\theta, \phi) = f(\theta, \phi) \tag{3.24}
\]

3. Using spherical harmonics, we express each function \(f_r\) as a sum of its different frequencies:

\[
f_r(\theta, \phi) = \sum_{l=0}^{\infty} f_r^l(\theta, \phi) \tag{3.25}
\]

\[
f_r^l(\theta, \phi) = \sum_{m=-l}^{l} a_{lm} Y_l^m(\cos \theta)e^{im\phi} \tag{3.26}
\]

where \(a_{lm}\) are constants and \(Y_l^m(\theta, \phi)\) Laplace’s spherical harmonics basis function of degree \(l\) and order \(m\).

4. Noting that rotations do not change the \(L_2\) norm of functions, i.e., \(\| f_r^l \|\) does not change if we rotate the function \(f_r\). We define a rotation invariant signature for \(f_r\) as the collection of scalars \(\| f_r^0 \|, \| f_r^1 \|, \ldots\). In practice, we can only store finite frequency components, so a bandwidth \(N_L\) is set to eliminate those beyond the bandwidth. The signature for \(f_r\) becomes \(\| f_r^l \|_{N_L}\).

5. Combining these different signatures over the different radii, we obtain a 2D histogram \(H(\mathcal{F}) = [H_r]_{N_R \times N_L}\),

\[
H_r(\mathcal{F}) = \| f_r^l \| \tag{3.27}
\]
3.2 Static Similarity Metrics

This gives a two-dimensional rotation invariant \textit{spherical harmonics descriptor},

\[ SHR(M) = H(F) \]  

Figure 3.4 illustrates the SHR functions and coefficient distribution for the single 3D video frame of a person.

\section{Static Similarity Metrics}

The shape descriptors presented in the previous section can be used to define a similarity measure. Given two individual 3D video frames \( p \) and \( q \) with reconstructed 3D meshes \( M_p \) and \( M_q \), the shape similarity \( s(p, q) \) is then defined as follows:

For \textit{Shape Distribution},

\[ s(p, q) = |SD(M_p) - SD(M_q)| \]  

For \textit{Spin Image},

\[ s(p, q) = |SI(M_p) - SI(M_q)| \]  

For \textit{Spherical Harmonic Representation},

\[ s(p, q) = |SHR(M_p) - SHR(M_q)| \]  

where \( | \cdot | \) compute the Euclidean distance. However, Shape Histogram is not rotation invariant, we have to re-define \( s(p, q) \) to take rotation into account. Here, we assume human models have an upright direction, since we consider a human pose laying on
the ground to be different from a standing pose, even though their shapes are similar since they cannot be concatenated seamlessly. The shape similarity $s(p, q)$ for Shape Histogram is then defined as follows,

$$ s(p, q) = \min_{\phi} |SH(M_p) - SH(R(M_q, \phi))| $$

(3.32)

where $R(M_q, \phi)$ denotes the 3D mesh $M_q$ is rotated by $\phi$ around the vertical axis. In practice, instead of rotating the 3D mesh model $M_p$ we first construct a high-resolution Shape Histogram $SH^*(M_q)$ and store it. Computing the minimal similarity against different $\phi$ requires shifting $SH^*(M_q)$ in dimension $\phi$ and rebinning back to $SH(M_q)$ for comparison. Note that subsampling a high resolution signal without low-pass filtering may cause aliasing. This process of rebinning (taking sum of high resolution bins) is very similar to applying a low-pass filter over the high resolutional signal, therefore, avoid aliasing problem. Since we only consider the rotation about the vertical axis, the bin size of high-resolution Shape Histogram is set to only increase the resolution in dimension $\phi$,

$$ (\Delta r^*, \Delta \theta^*, \Delta \phi^*) = (\Delta r, \Delta \theta, 1^\circ) $$

(3.33)

Therefore, we can compute the minima by shifting the histogram with an array $\phi_n = [0, 1, .., 359]$,

$$ s(p, q) = \min_{\phi_n} |SH(M_p) - B(SH^*(M_q, \phi_n))| $$

(3.34)

where $B(\cdot)$ denotes rebinning the high-resolution Shape Histogram $SH^*(\cdot)$ back to $SH(\cdot)$ and $SH^*(M_q, \phi_n)$ shifting $SH^*(M_q)$ with $\phi_n$ bins in the dimension of $\phi$. This evaluates all possible rotation around the vertical axis with $1^\circ$ resolution. Although the shifting of high-resolution Shape Histograms may suffer aliasing, after re-binning to low-resolution Shape Histograms for comparison, the effect of aliasing will be reduced and then not affect the similarity measure or recognition performance much.

Finally, we can define a static similarity matrix. Given two 3D video sequences, $P = \{p_i\}_{N_p}$ and $Q = \{q_j\}_{N_Q}$, the frame-to-frame similarity matrix $S$ is defined,

$$ S := (s_{ij})_{N_p \times N_Q} $$

(3.35)

$$ s_{ij} = s(p_i, q_j) $$

(3.36)
3.3 Performance Evaluation

The performance of the shape descriptors is evaluated using a ground-truth data-set from simulated data. Temporal mesh sequences are constructed for different motions and the classification of correct and incorrect similarity is assessed against ground-truth using the Receiver-Operator Characteristic (ROC) curves for each technique. Optimal parameter settings for each technique are first determined by their ROC performance. Descriptors with optimal parameter setting are then compared. Finally, the best descriptor in terms of ROC performance is used to identify similar frames in real 3D video sequences.

3.3.1 Ground Truth

A simulated data-set is created using 14 articulated character model for people animation using 28 motion capture sequences. Animated models of people with different gender, body-shape and clothing were reconstructed from multiple view images [99]. The height varies between about 1.6m to 1.9m. Each model has a single surface mesh with 1K vertices and 2K triangles. Models were animated by using 28 skeleton motion capture sequences from the Santa Monica mocap archive for the following motions: sneak, walk (slow, fast, turn left/right, circle left/right, cool, cowboy, elderly, tired, macho, march, mickey, sexy, dainty), run (slow, fast, turn right/left, circle left/right), sprint, vogue, faint, rock n'roll, shoot. Each sequence comprised 100 frames giving a total of 39200 frames of synthetic 3D video with known ground-truth correspondence. Figure 3.5 shows 14 models and example frames of multiple motions for one model. Given the known correspondence rigid-body registration can be performed to align the frames for ground-truth assessment of similarity. The known correspondence is only used to compute the true ground-truth surface distance, and is not used in computing any of the shape similarity measures.

The ground-truth shape similarity between two surfaces is measured using the average distance between corresponding vertices. This characterises the frame-to-frame difference between the surfaces. Given two 3D video frames $p$ and $q$ with reconstructed 3D meshes $M_p = < V_p, F_p >$ and $M_q = < V_q, F_q >$ which have vertex-to-vertex correspondence, i.e. both $V_p = \{v^P_k\}$ and $V_q = \{v^q_k\}$ have $N_V$ vertices and the same index
the ground to be different from a standing pose, even though their shapes are similar since they cannot be concatenated seamlessly. The shape similarity \( s(p, q) \) for Shape Histogram is then defined as follows,

\[
s(p, q) = \min_{\phi} |SH(M_p) - SH(R(M_q, \phi))|
\]

where \( R(M_q, \phi) \) denotes the 3D mesh \( M_q \) is rotated by \( \phi \) around the vertical axis. In practice, instead of rotating the 3D mesh model \( M_p \) we first construct a high-resolution Shape Histogram \( SH^*(M_q) \) and store it. Computing the minimal similarity against different \( \phi \) requires shifting \( SH^*(M_q) \) in dimension \( \phi \) and rebinning back to \( SH(M_q) \) for comparison. Note that subsampling a high resolution signal without low-pass filtering may cause aliasing. This process of rebinning (taking sum of high resolution bins) is very similar to applying a low-pass filter over the high resolutional signal, therefore, avoid aliasing problem. Since we only consider the rotation about the vertical axis, the bin size of high-resolution Shape Histogram is set to only increase the resolution in dimension \( \phi \),

\[
(\Delta r^*, \Delta \theta^*, \Delta \phi^*) = (\Delta r, \Delta \theta, 1^\circ)
\]

Therefore, we can compute the minima by shifting the histogram with an array \( \phi_n = [0, 1, \ldots, 359] \),

\[
s(p, q) = \min_{\phi_n} |SH(M_p) - B(SH^*(M_q, \phi_n))|
\]

where \( B(\cdot) \) denotes rebinning the high-resolution Shape Histogram \( SH^*(\cdot) \) back to \( SH(\cdot) \) and \( SH^*(M_q, \phi_n) \) shifting \( SH^*(M_q) \) with \( \phi_n \) bins in the dimension of \( \phi \). This evaluates all possible rotation around the vertical axis with 1° resolution. Although the shifting of high-resolution Shape Histograms may suffer aliasing, after re-binning to low-resolution Shape Histograms for comparison, the effect of aliasing will be reduced and then not affect the similarity measure or recognition performance much.

Finally, we can define a static similarity matrix. Given two 3D video sequences, \( P = \{p_i\}_{N_P} \) and \( Q = \{q_j\}_{N_Q} \), the frame-to-frame similarity matrix \( S \) is defined,

\[
S := (s_{ij})_{N_P \times N_Q}
\]

\[
s_{ij} = s(p_i, q_j)
\]
3.3. Performance Evaluation

The performance of the shape descriptors is evaluated using a ground-truth data-set from simulated data. Temporal mesh sequences are constructed for different motions and the classification of correct and incorrect similarity is assessed against ground-truth using the Receiver-Operator Characteristic (ROC) curves for each technique. Optimal parameter settings for each technique are first determined by their ROC performance. Descriptors with optimal parameter setting are then compared. Finally, the best descriptor in terms of ROC performance is used to identify similar frames in real 3D video sequences.

3.3.1 Ground Truth

A simulated data-set is created using 14 articulated character model for people animation using 28 motion capture sequences. Animated models of people with different gender, body-shape and clothing were reconstructed from multiple view images [99]. The height varies between about 1.6m to 1.9m. Each model has a single surface mesh with 1K vertices and 2K triangles. Models were animated by using 28 skeleton motion capture sequences from the Santa Monica mocap archive for the following motions: sneak, walk (slow, fast, turn left/right, circle left/right, cool, cowboy, elderly, tired, macho, march, mickey, sexy, dainty), run (slow, fast, turn right/left, circle left/right), sprint, vogue, faint, rock n'roll, shoot. Each sequence comprised 100 frames giving a total of 39200 frames of synthetic 3D video with known ground-truth correspondence. Figure 3.5 shows 14 models and example frames of multiple motions for one model. Given the known correspondence rigid-body registration can be performed to align the frames for ground-truth assessment of similarity. The known correspondence is only used to compute the true ground-truth surface distance, and is not used in computing any of the shape similarity measures.

The ground-truth shape similarity between two surfaces is measured using the average distance between corresponding vertices. This characterises the frame-to-frame difference between the surfaces. Given two 3D video frames $p$ and $q$ with reconstructed 3D meshes $M_p = < V_p, F_p >$ and $M_q = < V_q, F_q >$ which have vertex-to-vertex correspondence, i.e. both $V_p = \{ v^n_p \}$ and $V_q = \{ v^n_q \}$ have $N_V$ vertices and the same index
Figure 3.5: Synthetic Dataset. (a) 3D models (top left to bottom right): Adrian, Alan, Dave, EngJon, Graham, Jez, Jigna, Joel, Marc, PengWei, Pete, Pip, Venura, Yacob; (b) Example frames from different motion sequences of Jigna in the Synthetic Dataset (top to bottom): Sneak, Fast Walk, Sprint, Walk Mickey, Rock&Roll, Vogue Dance.
3.3. Performance Evaluation

\[ d_P(p, q) = \frac{1}{N_V} \sum_{k=0}^{N_V-1} |v_k^p - v_k^q| \]  
\[ d_V(p, q) = \frac{1}{N_V} \sum_{k=0}^{N_V-1} |u_k^p - u_k^q| \]

where \( u_k^P = v_k^P(t+1) - v_k^P(t) \), \( v_k^Q = v_k^Q(t+1) - v_k^Q(t) \) are velocity vectors, \( t+1, t \) denote the next and current frame. Given two 3D video sequences \( P = \{p_i\}_{N_P} \) and \( Q = \{q_j\}_{N_Q} \), ground-truth similarity is defined as a combination of vertex-to-vertex position and velocity similarity,

\[ S^{GT} := (s_{ij}^{GT})_{N_P \times N_Q} \]
\[ s_{ij}^{GT} = (1 - \omega) \cdot d_P(p_i, q_j) + \omega \cdot d_V(p_i, q_j) \]

Throughout the results presented in the thesis \( \omega \) is set as 0.5 to balance the vertex position and velocity similarity. For classification of frames as similar a threshold \( \tau \) is set on the ground-truth similarity where the similarity falls below a fixed predefined threshold \( \tau^{GT} \). After normalisation of the similarity to the range \([0, 1]\) the similarity threshold is set to \( \tau^{GT} = 0.3 \) throughout this work which gives the ground-truth binary classification matrix \( C^{GT} \),

\[ C^{GT} := (c_{ij}^{GT})_{N_P \times N_Q} \]
\[ c_{ij}^{GT} = \begin{cases} 1 & \text{if } s_{ij}^{GT} < \tau^{GT} \\ 0 & \text{otherwise} \end{cases} \]

A ground-truth binary classification classify frames \( p_i \) and \( q_j \) as similar \( (c_{ij}^{GT} = 1) \) and dissimilar \( (c_{ij}^{GT} = 0) \) otherwise. Inclusion of the surface motion in the ground-truth similarity together with the shape removes the ambiguity inherent in static single frame similarity measures. The lines of similarity in the diagonal direction indicate the periodic structure of the synthetic 3D video sequences. Figure 3.6 shows the similarity and ground-truth classification for the 28 motions with one of the models.
Figure 3.6: Ground truth self-similarity and classification of 28 motions from "Jigna" (from top left to bottom right in order): sneak, slow walk, fast walk, slow run, fast run, sprint, walk circle (left and right), run circle (left and right), walk turn (left and right), run turn (left and right), walk in styles (cool, cowboy, dainty, elderly, macho, march, mickey, sexy, tired, toddler) and complex motions (rock and roll, vogue dance, faint, shot arm).
3.3. Performance Evaluation

3.3.2 Evaluation Criterion

Performance of the shape descriptors is evaluated using the ROC curve, showing the true-positive rate (TPR) or sensitivity in correctly defining similarity against the false-positive rate (FPR) or one-specificity where similarity is incorrect.

\[
TPR = \frac{ts}{ts + fd} \quad (3.43)
\]

\[
FPR = \frac{fs}{fs + td} \quad (3.44)
\]

where \( ts \) denotes the number of true-similar predictions, \( fs \) the false similar, \( td \) true dissimilar and \( fd \) false dissimilar in comparing the predicted similarity between two frames to the ground-truth similarity.

The similarity for each shape descriptor is normalised to the range \( 0 \leq s'_{ij} \leq 1 \).

\[
s'_{ij} = \frac{s_{ij} - s_{\text{min}}}{s_{\text{max}} - s_{\text{min}}} \quad (3.45)
\]

where \( s_{\text{min}} = 0 \) and \( s_{\text{max}} \) is the maximal dissimilarity over all \( s_{ij} \in S \) similarity matrix of the whole database.

A binary classification matrix for the shape descriptor \( C(\tau) = [c_{ij}(\tau)] \in \{0, 1\} \) is then defined

\[
c_{ij}(\tau) = \begin{cases} 
1 & \text{if } s'_{ij} < \tau \\
0 & \text{otherwise}
\end{cases} \quad (3.46)
\]

The classification \( c_{ij}(\tau) \) for a given \( \tau \) is then compared to the ground-truth similarity classification \( c_{ij}^{\text{GT}} \) defined in section 3.3.1. The number of true and false similarity classifications is then counted:

\[
\begin{align*}
ts(\tau) &= \sum_{ij} c_{ij}^{\text{GT}} \cdot c_{ij}(\tau) \quad (3.47) \\
td(\tau) &= \sum_{ij} (1 - c_{ij}^{\text{GT}}) \cdot (1 - c_{ij}(\tau)) \quad (3.48) \\
fs(\tau) &= \sum_{ij} (1 - c_{ij}^{\text{GT}}) \cdot c_{ij}(\tau) \quad (3.49) \\
fd(\tau) &= \sum_{ij} c_{ij}^{\text{GT}} \cdot (1 - c_{ij}(\tau)) \quad (3.50)
\end{align*}
\]

The ROC performance for a given shape similarity measure is obtained by varying the threshold \( \tau \in [0, 1] \) to obtain the true TPR(\( \tau \)) and false FPR(\( \tau \)) positive rates according to equation 3.44.
3.3.3 Parameter Setting

Optimal parameter setting for each of the shape descriptors to match frames in 3D video sequences of people are determined by evaluating the ROC curve for a range of parameter settings using a synthetic human motion [52]. This evaluation uses the known ground-truth similarity to evaluate the classification performance. Table 3.3.3.4 presents the optimal parameters for each shape descriptor. These parameter settings are used throughout the evaluation.

3.3.3.1 Shape Distribution

Shape Distribution has two parameters, \( N_p \), the number of samples and \( N_d \), the number of fixed bins to use in constructing the \( D^2 \) distribution. Empirically, we set \( N_p = 10^6 \) and \( N_d = 10^3 \) which are reported with low enough variance and high enough resolution in [91]. Figure 3.1(a) shows the performance with different value of \( N_p \), as the number of samples increases the performance converges to a limit and at \( N_p = 10^6 \) there is no further increase in performance is apparent.

3.3.3.2 Spin Image

The Spin Image has two parameters, \( N_\alpha \), the number of bins for \( \alpha \), and \( N_\beta \), the number of bins for \( \beta \) to use in constructing the 2D histogram. The bins correspond to rectangular regions and an equal spatial resolution is adopted \( N_\alpha = N_\beta \). Figure 3.7(b) shows the performance with different value of \( N_\alpha \). A limit is found at \( N_\alpha = 40 \) where a subsequent increase in resolution reduces performance. This can be explained that as \( N_\alpha \) increases, although the resolution in re-sampling a surface is increased, surfaces that are spatially close will become re-sampled into different bins and so the overlap between the histograms is in turn reduced.

3.3.3.3 Shape Histogram

Six variants of the shape histogram are constructed. A shell histogram with surface and volume samples, and a spherical coordinate histogram with surface and volume samples using PCA alignment and by evaluating similarity up to a rotation. The shell histogram is defined by the number of shells \( N_r \). The spherical coordinate histogram
by the number shells $N_r$, and angular bins $N_\phi, N_\theta$. The resolution of the angular bins is set equal at $N_\phi = 2N_\theta$. Figure 3.7(d)-3.7(i) shows them with different configuration of $N_r$ and $N_\theta$.

It can be seen that the spherical-coordinate histogram with PCA alignment performs least-well. This is a result of the ambiguity in consistently defining the principal axes used to align two surfaces. As an object becomes more symmetric, the eigenvalues approach making the definition of a consistent set of axes ambiguous. The surface-shell histogram outperforms the volume-shell histogram and is equivalent to the spin-image descriptor with the loss of one-degree of freedom. The highest performance is obtained using the rotated spherical-coordinate histogram which introduces a third degree-of-freedom in the descriptor and derives the best alignment between two surfaces to evaluate similarity. It is interesting to note that performance is improved by using a relatively coarse histogram comparison and the volume-descriptor is more robust to changes in the parameters $N_r, N_\theta, N_\phi$. The spherical volume-descriptor is adopted with bins $N_r = 5, N_\theta = 10, N_\phi = 20$.

### 3.3.3.4 Spherical Harmonics

The spherical harmonic descriptor is controlled by two parameters $N_R$, the number of shells and $N_L$, the bandwidth, that is the range of frequencies preserved in the harmonics. The greater the value of $N_R$ and $N_L$ the higher the resolution of the descriptor. For simplicity we set $N_R = N_L$. Figure 3.7(c) shows that performance is relatively stable to different resolution. We choose $N_R = 30$ which provides adequate granularity for discriminating shapes while filtering out high-frequency noise in the original data suggested in [37].

### 3.3.4 Evaluation of 3D Shape Descriptors

Similarity curves at the first frame in the sprint motion are shown in Figure 3.8. The near-optimal parameter settings for the descriptors selected in the run-circle motion are adopted. The results demonstrate that the best performance is obtained using the spherical-coordinate histogram technique. The performance of Spin Image drops when applied to the sprint motion. This may be explained by an inherent reflective
Table 3.1: Parameter settings for shape descriptors [52].

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shape Distribution</td>
<td>No. of samples $N_p = 10^6$, $N_d = 10^3$</td>
</tr>
<tr>
<td>Spin Image</td>
<td>No. of bins $N_\alpha = N_\beta = 40$</td>
</tr>
<tr>
<td>Shape Histogram</td>
<td>No. of radial bins $N_r = 5$</td>
</tr>
<tr>
<td></td>
<td>No. of angular bins $N_\theta = 10$, $N_\phi = 20$</td>
</tr>
<tr>
<td>Spherical Harmonics</td>
<td>No. of radial shells $N_R = 30$</td>
</tr>
<tr>
<td></td>
<td>No. of harmonics (bandwidth) $N_L = 30$</td>
</tr>
</tbody>
</table>

ambiguity in the descriptor: an object and its mirror have the same descriptor. For example, in Figure 3.8, frame 29 is quite dissimilar to frame 1 (query frame) but very similar to its mirror, due to the reflective ambiguity, it obtains a valley in the similarity curve. The Shape Distribution descriptor is also invariant to a mirror transformation [91] and has a relatively coarse discrimination. The poorest performance is obtained using spherical harmonics. This is a surprising result considering the performance demonstrated in the 3D shape retrieval literature. This may be explained in part by the inherent “information loss” in the descriptor [62]: the descriptor is unchanged if we apply different rotations to the different frequency components of a spherical function; the harmonic representation only stores the energy in each frequency component which does not provide enough information to reconstruct the orientation of the 3D shape. This means dissimilar shapes may have the same descriptors, the descriptor is therefore more likely to produce false-positive shape matches. For example, in Figure 3.8, frame 14 is quite dissimilar to frame 1 but frames 14-20 all form a local minima in the similarity curve. We conclude that for the problem of matching articulated human motion, the spherical volume-histogram descriptors provide the best performance. The descriptors are also demonstrated to be relatively insensitive to parameter setting as shown in Figures 3.7(i).

However, frame-to-frame similarity does not take into account the motion dynamics, and therefore, suffers temporal ambiguity. Figure 3.9(a) shows the self-similarity and classification evaluated using the known ground truth surface correspondence for shape and motion (section 3.3.1). The periodic structure of the walking motion is illustrated...
by the diagonal lines of high similarity (dark blue). The self-similarity matrix for the four static shape descriptors computed independently frame-to-frame without known correspondence is shown in Figure 3.9(b-e). The static shape similarity for all descriptors gives high similarity (dark blue) diagonal lines corresponding to the ground-truth periodic motion structure. Additional lines of high-similarity occur due to ambiguities in the static shape descriptor. Anti-diagonal lines of high-similarity occur with all static shape descriptors due to frames with similar shape but opposing motion such as the mid-point of the walk cycle (marked as a triangle) illustrated in Figure 3.9(g). For the Shape Distribution and Spherical Harmonic similarity measures there is also a periodic line structure of high-similarity at twice the motion frequency in the diagonal direction due to mirror ambiguity where a shape and its mirror image have the same descriptor. An example of this (marked as a circle) is illustrated in Figure 3.9(f) where frames 47—51 are dissimilar to frames 66—70 but similar to their mirror image.

3.4 Similarity Measure on Real Data

Finally, we apply the rotated volume-spherical Shape Histogram (SHvr) descriptor to a captured 3D surface sequences of a street-dancer from a public database [102]. The Real Dataset includes a street dancer (JP) performing eight motions with baggy clothing. Each motion contains 250 to 500 frames and in each frame a 3D mesh contains around 140k vertices and 280k triangles. Although the surface resolution of real data is much higher than synthetic data, the evaluation on synthetic dataset and real data is still compatible, this is because that we construct Shape Histograms from voxels instead of vertices and after voxelization in the same voxel resolution the difference of surface resolution is much reduced and can be ignored without affecting the recognition performance. The motion is fast and complex with the topology changing frequently, which is very difficult to deal with by applying traditional skeleton-based similarity metrics. In addition, each 3D mesh is reconstructed from multi-view image independently, hence, there is no hierarchy information or temporal surface correspondence, which increases the difficulty to deal with the data. Figure 3.10 shows some example frames from these motion sequences of real 3D video. Similarity matrices produced by SHvr are shown in Figure 3.11 for intra-motion and
Chapter 3. 3D Shape Similarity

Figure 3.12 for inter-motion.

In Figure 3.11, self-similarity matrices for all motions correctly show a strong similarity (dark blue) along the diagonal, top-left and bottom-right corners, where the actor start and end the performance with a stand pose. The self-similarity matrix for "pop2lock" correctly show a relatively strong similarity (green) along the the diagonal direction, where actor walk for a while.

In Figure 3.12, cross-similarity matrices for all motions correctly show a strong similarity (dark blue) at top-left or bottom-right corners, where the actor start or end the performance with a stand pose. The cross-similarity matrix for "pop" to "lock2pop" correctly show a strong similarity (dark blue) along the the diagonal direction, where the transition between motions happens. The cross-similarity matrix for "pop" to "kickup" also correctly show a strong dissimilarity (red) in the middle of the matrix, where a kick up in the air is quite different to all frames in "pop".

Finally, for a detailed comparison of 3D video frames and illustrates the capability of retrieval similar poses of a moving person. Similar shapes are selected by looking for the valley of the similarity curve. Some results are shown in Figure 3.13(a) and Figure 3.13(b), respectively for intra-motion pop dancing and inter-motion between pop and lock dancing.

3.5 Summary

In this chapter, we have presented a quantitative evaluation of existing static 3D shape descriptors applied to the problem of finding 3D shape similarity in sequences of a temporally varying articulated freeform object, the surface of a clothed person. Global shape distribution descriptors were adopted from the literature to provide an object description and allow for changes in captured surface topology. The Shape Distribution [91], Spin Image [58], Shape Histogram [3] and Spherical Harmonic [62] descriptors were compared. Performance was evaluated using the Receiver-Operator Characteristic in classifying correct versus incorrect similarity against ground-truth using a dataset synthesised from a 3D character model animated using motion capture data.

The highest performance is obtained from volume-sampling shape-histogram descriptors. The descriptors also demonstrate relative insensitivity to parameter setting. Prin-
3.5. Summary

cipal component based alignment of surfaces demonstrated poor performance in comparison to testing similarity up to a rotation. Shape Distribution and Spin Images demonstrate an ambiguity to reflective symmetry. Spherical harmonics provide an ambiguous descriptor that greatly reduced the specificity in defining similarity. While the evaluation is not exhaustive due to the breadth of the literature on 3D shape similarity the work clearly demonstrates the advantage of volume sampling shape-histogram in the context of shape matching in temporal surface sequences of people.
Figure 3.7: Parameter Setting. (a) Shape Distribution (SD); (b) Spin Image (SI); (c) Spherical Harmonics (SHR); Shape Histogram: (d) surface-shell (SHss); (e) surface-spherical histogram with PCA-alignment (SHsp); (f) surface-spherical histogram with rotation (SHsr); (g) volume-shell (SHvs); (h) volume-spherical histogram with PCA-alignment (SHvp); (i) volume-spherical histogram with rotation (SHvr).
3.5. Summary

Figure 3.8: Comparison for Shape Descriptors of 1st frame with sequence for sprint motion. Points marked on the similarity curves indicate the frame shown below.
Figure 3.9: Static similarity measure for motion "Fast Walk" in a straight line compared with itself. Self-similarity and classification obtained by (a) Temporal Ground-Truth (TGT). Self-similarity and classification at $FPR = 5\%$ (section 3.3.2) obtained by (b) the rotated volume-sampling Shape Histogram (SHvr); (c) Shape Distribution (SD); (d) Spin Image (SI); (e) Spherical Harmonics Representation (SHR). Example frames show (f) sub-sequences around frames 49 and 68 (centre) with "mirror ambiguity"; (g) sub-sequences around frames 24 and 40 (centre) with similar shape but different direction of motion for arms and legs.
3.5. Summary

Figure 3.10: Example frames from motion sequences of JP in the Real Dataset (top to bottom): Lock, Lock2Pop, Pop Dance.
Figure 3.11: Self-similarity matrices produced by SHvr of 8 motions from Real Dataset. JP in baggy clothing: (a) flashkick, free, head, kickup; (b) lock2pop, lock, pop2lock, pop. Note that the time direction is from top-left to bottom-right.
Figure 3.11: Self-similarity matrices produced by SHvr of 8 motions from Real Dataset. JP in baggy clothing: (a) flashkick, free, head, kickup; (b) lock2pop, lock, pop2lock, pop. Note that the time direction is from top-left to bottom-right.
Figure 3.12: Cross-similarity matrices produced by SHvr of pairs of motions from Real Dataset (from top left to bottom right in order). JP in baggy clothing: pop→kickup, pop→lock2pop, pop→lock, pop→pop2lock. Note that the time direction is from top-left to bottom-right, the “from” motion is along vertical axis and “to” motion along horizontal axis.
Figure 3.13: Similarity Measure on Real Data. Intra-motion of pop: (a) similarity curve of frame 1 with sequence. Inter-motion between pop and lock: (b) similarity curve of frame 139 of pop with lock sequence.
Chapter 4

Temporal Similarity

Neumann et al. [87] note that actions are best defined as 4D patterns in space and time, and similarity should ideally be compared in spatio-temporal space. However, current matching techniques from the shape retrieval literature consider static shapes only [106, 14, 36, 55]. Shapes at a single frame can appear similar, but over time can belong to very different actions. For example, if a pendulum swinging from left to right is split into several frames, each one may be easily confused with a right to left swing. In such a case, temporal information must be added to resolve the ambiguity. Temporal shape matching is a natural extension of current shape matching work. In this chapter, we focus on extending static similarity metrics including shape-only and shape-colour similarity to the time domain. A time-filter is first used to compute temporal similarity from existing static similarity. Then two different shape-flow descriptors are introduced to be more distinctive by also considering the motion trajectory. Finally, a quantitative evaluation against a simulated data-set is provided and the best performer is used to measure similarity for real 3D videos to find frames similar in shape and motion.

4.1 Temporal Descriptors and Similarity Metrics

In this section, we first extend static 3D shape-only and shape-colour descriptors to the time domain. Firstly, using three approaches by applying temporal filtering of the shape similarity matrix. Two novel temporal descriptors are proposed based on shape-flow of the shape histogram descriptor with global and local alignment of frames. Subsequent
Chapter 4. Temporal Similarity

Figure 4.1: Temporal similarity measure for motion “Fast Walk” in a straight line compared with itself. Self-similarity with window size 3, 5, 7, 9 and classification with window size 9 at $FPR = 5\%$ (section 3.3.2) obtained by (a) SHvrG; (b) SHvrS; (c) SHvrT; (d) SDT; (e) SIT; (f) SHRT.
4.1. Temporal Descriptors and Similarity Metrics

Performance evaluation of static and temporal shape descriptors against ground-truth and real 3D video sequences is presented in sections 4.2 and 4.3.

4.1.1 Time-filtered Descriptors

Temporal information can be incorporated in a static 3D shape-only or shape-colour descriptor using a time filter. Schödl et al. [96] and Efros et al. [29] use a similar strategy to achieve motion-to-motion matching in 2D video. In practice, the time filter is applied to the frame-to-frame similarity matrix obtained by 3D Shape Descriptors. The temporal filtered similarity matrix is obtained by convolving it with a time filter. Here, we use an average filtering. For time-filtered Spherical Harmonics Representation (SHRT), time-filtered Spin Image (SIT), time-filtered Shape Distribution (SDT). The time filtered similarity is evaluated as

$$ s_{ij}^t = \frac{1}{2N_t + 1} \sum_{k=-N_t}^{N_t} s_{i(\pm k)(j+\pm k)} $$

(4.1)

where a time filter with window size $2N_t + 1$. The computational complexity of the time-filtered shape descriptor is the cost of computing the frame-to-frame static shape similarity together with a convolution of the resulting similarity matrix with the temporal filter. This cost is dominated by the cost of computing the static shape similarity with a relatively small additional cost of time filtering.

Time-filtering emphasises the diagonal structure of the similarity matrix and reduces minima in the anti-diagonal direction resulting from motion and mirror ambiguities in the static shape descriptor [53]. Figure 4.1(c-f) illustrate the effect of time-filtering with increasing temporal window size for each of the shape descriptors on a periodic walking motion. Comparison with the temporal ground-truth Figure 3.9(a) shows that the incorrect shape similarity in the anti-diagonal direction which occur with static shape similarity is reduced. Performance evaluation of the time-filtered shape descriptors is presented in section 4.2.1.

4.1.2 Shape-flow Descriptors

In previous work, Weinland et al. [115] introduced Motion History Volumes (MHV) as a free-viewpoint representation for human actions recognition. The MHV fuses
visual hulls at each time instant over a short temporal window into a single three dimensional representation. Instead of using a single representation for a short time of action, two novel temporal shape descriptors are introduced to measure the change in shape for a surface in a sub-sequence corresponding to a given temporal window. The benefit is that the optimal temporal window is not necessarily pre-defined and can be found by applying an adaptive search scheme. Time filtering of the static shape similarity matrix does not enforce the temporal consistency in a motion as each static comparison is aligned independently on a frame-by-frame basis as illustrated in Figure 4.2(b). The new descriptors consider not only the similarity between individual frames in a sub-sequence but also preserve the temporal changes using a sub-sequence alignment, referred to as a shape-flow descriptor.

Performance evaluation of static shape descriptors in section 3.3.4 demonstrates that the volume sampling 3D spherical histogram gives the best performance in classifying shape similarity. The 3D histogram is extended here to incorporate changes in shape using a 4D histogram in which each 3D spherical bin records the shape over a 1D temporal window. Similarity is again defined using the $L_2$ distance between coarse histograms after alignment.

Histogram alignment is considered using two methods, either by finding the optimal alignment of the 3D descriptor for the centre frame of the temporal window, or by finding the optimal alignment of the entire sequence in the 4D temporal descriptor. We call the first method local shape-flow matching and the second global shape-flow matching. Figure 4.2(c,d) illustrate the alignment used in the local and global shape-flow similarity. Local shape-flow matching has the same computational complexity as static shape matching but may not find the optimal alignment for the whole sub-sequence. Global shape-flow matching is more robust but the computational cost is proportional to the time window size. Comparative performance evaluation of the temporal shape-flow descriptors is presented in section 4.2.2.

Optimal alignment is derived first by finding the translation that matches the centre of mass and then by shifting the histogram \(^1\) to give the greatest similarity. Here,

\(^1\)Shifting a SHvr descriptor in its $\phi$ bins is equivalent to rotating a mesh around the vertical axis but is more efficient (section 3.1.3)
4.1. Temporal Descriptors and Similarity Metrics

Figure 4.2: Shape-flow matching. Sequences (a) before applying any alignment; (b) after applying independent alignment as in time filtering for each frame used in static shape similarity; (c) after local shape-flow matching; (d) after global shape-flow matching.

an upright orientation is assumed. Let $SH(M_{pi})$ and $SH(M_{qj})$ denote 3D shape histograms of individual frames in two 3D video sequences $P = \{p_i\}$ and $Q = \{q_j\}$. The dissimilarity for a local shape-flow matching $s_{ij}$ is computed as follows,

$$s_{ij}^l = \frac{1}{2N_t + 1} \sum_{k=-N_t}^{k=N_t} |SH(M_{pi+k}) - SH(R(M_{qj+k}, \phi_{opt}))|$$  (4.2)

$$\phi_{opt} = \arg \min_{\phi_n} |SH(M_{pi+k}) - SH(R(M_{qj+k}, \phi_n))|$$  (4.3)

where $R(M_{qj+k}, \phi_{opt})$ denotes the 3D mesh $M_{qj+k}$ is rotated by $\phi_{opt}$ around the vertical axis. $\phi_{opt}$ is optimal rotation around vertical axis which best align two compared 3D mesh. $\phi_n = [0, ..., 359]$.

For global shape-flow matching, the optimal rotation is found by searching for the rotation that minimises the distance between two sub-sequences and the similarity matrix is computed as follows:

$$s_{ij}^g = \min_{\phi_n} \left\{ \frac{1}{2N_t + 1} \sum_{k=-N_t}^{k=N_t} |SH(M_{pi+k}) - SH(R(M_{qj+k}, \phi_n))| \right\}$$  (4.4)

The effect of the shape-flow descriptors for a walking motion is illustrated in Figure 4.1 and Figure 4.3. For motion in a straight line, Figure 4.1(a—c) show that similar results are obtained for shape-flow with local and global frame alignment and the temporal filtering with independent alignment of frames. Comparison of the shape-flow descriptors ($SH_{rvG}$, $SH_{rvS}$) and the time-filtered descriptor ($SH_{rvT}$) based on shape histograms with other time filtered descriptors ($SDT, SIT, SHRT$) are shown in Figure 4.1(d—f). This demonstrates reduced temporal ambiguity with similarity and
Figure 4.3: Temporal similarity measure for motion "Fast Walk" in a straight line compared with motion "Fast Walk" on a spiral. Cross-similarity and classification obtained by (a) Temporal ground-truth (TGT). Cross-similarity with window size 9 and classification at $FPR = 5\%$ (section 3.3.2) obtained by (a) Temporal ground-truth (TGT); (b) Global shape-flow alignment volume-sampling spherical Shape Histogram (SHvrG); (c) Local shape-flow alignment volume-sampling spherical Shape Histogram (SHvrS); (d) Temporal filtered volume-sampling spherical Shape Histogram (SHvrT); (e) Temporal filtered Spherical Harmonics Representation (SHRT).
4.2 Performance Evaluation

classification close to ground-truth which is shown in 3.9(a). Distinction between the performance of shape-flow and time-filtered descriptors can be seen when the motion is not in a straight line. Figure 4.3 illustrates the cross-similarity between walking in a straight line and on a spiral for the ground-truth and temporal shape descriptors. Figure 4.3(d,e) show that time-filtered descriptors SHvrT and SHRT fail to correctly characterise the change in similarity due to the non-linear motion path (e.g. spiral) resulting in changes in direction between consecutive frames. Figure 4.3(b,c) shows that the shape-flow descriptors SHvrG with global and SHvrS with local alignment of frames produce similarity and classification which closely match the ground-truth Figure 4.3(a). This illustrates a limitation of temporal filtering in correctly estimating similarity for non-linear motion paths which usually implies non-linear change of motion direction and shows that shape-flow descriptors overcome this limitation. Quantitative performance evaluation is presented in section 4.2. Figure 4.4 illustrates a limitation of shape-flow with local single frame alignment SHvrS versus global multiple frame alignment SHvrG for cross-similarity between real 3D video sequences of walk and jog motions. For shape-flow with global frame alignment SHvrG similarity Figure 4.4(a) the diagonal structure is clearly visible. However, with local frame alignment Figure 4.4(b) incorrect low-similarity scores occur on the diagonal (marked with a circle), this is due to failure of the local shape-flow alignment. Errors occur in SHvrS due to incorrect estimation of the alignment at the central frame. SHvrG is robust as optimal alignment is estimated for a sequence of frames.

4.2 Performance Evaluation

Performance of the shape descriptors is evaluated using a ground-truth dataset from synthetic data that is summarised in Table 4.1. Temporal mesh sequences are constructed for different motions and the classification of correct and incorrect similarity is assessed using the Receiver-Operator Characteristic (ROC) curves for each technique. This evaluation extends previous comparison of shape descriptors for a single person performing eight motions [52, 53] to a comprehensive dataset comprising models of 14 people each performing 28 motions, 39200 frames in total. Optimal parameter settings for each shape descriptor are determined by evaluating the ROC for different parameter
Chapter 4. Temporal Similarity

Figure 4.4: An example of the failure of SHvrS. Rachel’s “Jog” compared to “Walk” with a fixed window size 9 ($N_t = 4$) using (a) global shape-flow SHvrG; (b) local shape-flow SHvrS; (c) time-filtering shape histogram SHvrT.

settings [52]. Similarity measures are then evaluated against temporal ground-truth to identify similar frames in the 3D video sequences. The parameter setting, ground-truth and evaluation criterion remain the same as Section 3.3.3, Section 3.3.1 and Section 3.3.2 respectively in previous chapter.

We evaluate temporal shape descriptors defined in section 4.1 against the Temporal Ground Truth. Optimal parameter settings for the shape descriptors given in Table 3.3.3.4 are used for evaluation of the temporal descriptors. Since the optimal temporal window size will depend on the rate of motion with a larger window size being required for slow motions, performance of each of the temporal shape descriptors is evaluated for each motion using the ROC curve with a range of temporal window size.

4.2.1 Evaluation of time-filtered descriptors

Combined ROC curves of the time-filtered descriptors for evaluating self-similarity against temporal ground truth across all people and motions in the simulated dataset with an increasing temporal window size are shown in Figure 4.5(b-j). The performance of all descriptors increases compared to the equivalent static shape similarity in Figure 4.5(a). The time-filtered volume-sampled shape histogram SHvrT gives the highest performance of all time-filtered shape descriptors against temporal ground-truth. This is expected as the volume-sampled shape histogram SHvr gives the best performance for the static shape descriptors and time-filtering reduces the temporal ambiguity in-
4.2. Performance Evaluation

Figure 4.5: Evaluation of ROC curves for static and time-filtered descriptors on self-similarity across 14 people each performing 28 motions.
Performers | Motions per person | \( N_s \) | \( N_f \)  
--- | --- | --- | ---  
Adrian, Alan, Dave, | speed: sneak, walk/run (slow, fast), sprint | 6 | 600  
EngJon, Graham, | direction: walk/run (turn/circle left/right) | 8 | 800  
Jez, Jigna, Joel, | style: walk (cool, cowboy, dainty, elderly, | 10 | 1000  
Marc,PengWei, Pete, | macho, march, mickey, sexy, tired, toddler) |  |  
Pip, Venura, | dance: rock-and-roll, vogue | 2 | 200  
Yacob | non-periodic: faint, shot-arm | 2 | 200  
Sub-total per person: | | 28 | 2800  
Total: | | 392 | 39200  

Table 4.1: Synthetic 3D Video datasets for 14 people each performing 28 motions. \( N_s \) is the number of sequences and \( N_f \) the number of frames.

Increasing the classification accuracy. Comparison of the different shape descriptors with respect to window size also shows that the time-filtered shape histogram \( \text{SHvrT} \) is relatively insensitive to the change of window size. Figure A.2 in Appendix A gives a more detailed comparison by presenting individual ROC curves for the 28 motions with 14 people for each of the time-filtered shape descriptors with a fixed temporal window size. This demonstrates that the time-filtered volume-sampled shape histogram \( \text{SHvrT} \) has low inter-person variance and consistently outperforms other time-filtered shape descriptors.

### 4.2.2 Evaluation of shape-flow descriptors

Combined ROC curves of the shape-flow descriptors for classification of self-similarity against temporal ground truth across all people and motions in the simulated data set are presented in Figure 4.5(b–j) with increasing temporal window size. Characteristics for the shape-flow descriptors \( \text{SHvrG} \) and \( \text{SHvrS} \) are superimposed with the time-filtered descriptor \( \text{SHvrT} \) showing that the performance is similar for straight line motions. Analysis of the detailed characteristics shows that in general the global shape-flow \( \text{SHvrG} \) achieves the highest performance with time-filtering \( \text{SHvrT} \) and local shape-flow \( \text{SHvrS} \) marginally lower. The difference between aggregate characteristics is lower than the variance for different people and motions as shown in Figure A.3.
4.2. Performance Evaluation

Figure 4.6: Evaluation of ROC curves for Shape-flow descriptors on cross-similarity "Fast Walk" in a straight line and on a spiral across 14 people.

Figure 4.6 shows a case when shape-flow descriptor SHvrG and SHvrS achieve significantly higher performance than time filtering SHvrT for motion on a non-linear path. Shape-flow has significantly better performance than time-filtering for all window sizes. This is because that the independent alignment used by time-filtered descriptors does not take into account changes in motion direction between consecutive frames which resulting more incorrectly similarity matching and hence increases the false positive rate. ROC curves for local and global shape-flow in Figure 4.6 are superimposed indicating that performance of local and global frame alignment is comparable. This could be explained that although the local shape-flow cannot guarantee correctly characterising the change in similarity due to the change in direction between consecutive frames 4.1.2, in most cases, the local frame alignment is very similar to the global frame alignment and hence local shape-flow tends to give a comparable performance with global shape-flow. Table 4.2 presents the relative computational cost for time-filtering and shape-flow measures with increasing window size. This shows that there
is an order of magnitude increase in computational cost for the shape-flow descriptor with global frame alignment SHvrG, whereas local frame shape-flow and time-filtering have a computational cost similar to the static shape descriptor \( N_t = 0 \). In conclusion, the global shape-flow descriptor (SHvrG) overcomes the limitation of temporally filtered static shape descriptors (SHvrT,SIT,SDT,SHRT) for 3D video sequences with non-linear motion paths and is robust to errors in local frame alignment which occur with (SHvrS).

### 4.3 Similarity Measure on Real Data

In this section we apply the time-filtering shape histograms SHvrT to captured 3D video sequences of people. Real 3D video sequences were reconstructed from multiple camera video capture available as a public research database [102]. The real 3D video sequences of nine people performing a variety of motion used in this evaluation are summarised in Table 4.3. These include a street dancer (JP) performing complex movements with baggy clothing, a performer (Roxanne) wearing 3 different costumes with shorts, a short-dress and a long-dress together with seven other actors performing a standard set of movements. Captured 3D video sequences are unstructured meshes with unknown temporal correspondence and time varying mesh connectivity, topology and geometry.

Time-filtering shape histograms SHvrT are used to evaluate intra-person similarity between the 3D video sequences of different motions for each performer/costume combination and the inter-person similarity for different performers performing the same motion. Evaluation has been performed for all available sequences. Example results are
4.3. Similarity Measure on Real Data

<table>
<thead>
<tr>
<th>Performer</th>
<th>Motions</th>
<th>$N_s$</th>
<th>$N_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>JP</td>
<td>street dance: lock, pop, flash-kick, free-dance, head-spin, kickup + transitions</td>
<td>8</td>
<td>2300</td>
</tr>
<tr>
<td>Roxanne</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Game Character</td>
<td>walk, jog, stand, stagger, hit, tense + transitions</td>
<td>10</td>
<td>442</td>
</tr>
<tr>
<td>- Fashion 1</td>
<td>walk, pose, twirl + transitions</td>
<td>6</td>
<td>491</td>
</tr>
<tr>
<td>- Fashion 2</td>
<td>walk, pose, twirl + transitions</td>
<td>6</td>
<td>435</td>
</tr>
<tr>
<td>Others</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Adrian Gordon</td>
<td>idle, walk, jog, kick, punch + transitions</td>
<td>24</td>
<td>875</td>
</tr>
<tr>
<td>Gregor Rachel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jon Rafael Tony</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td></td>
<td>54</td>
<td>4543</td>
</tr>
</tbody>
</table>

Table 4.3: Real 3D Video datasets for 9 actors and motions (transition motions are sequences which transition from one motion to another i.e. walk to jog) $N_s$ is the number of sequences and $N_f$ the number of frames.

presented demonstrating typical results with identification of frames with similar shape and motion. SHvrT is applied to all sequences with the optimal resolution parameters (Table 3.3.3.4) and a temporal window size of 9 ($N_t = 4$). Intra-person similarity across different motions for several performers together with an example similarity curve are presented in Figure 4.7. The example matched frames for each performer show that the temporal similarity metric identifies frames of similar pose and motion across the different motions performed by each actor. In Figures 4.7(b,c) for Roxanne the similarity clearly identifies the periodic structure of the walking motion and identifies frames with similar shape and motion even with the highly non-rigid movement of the loose dress and long-hair. Figures 4.7(g,h) for the street dancer JP performing complex movements shows there is a lot of visible structure in the similarity matrix, frames with similar pose and motion are also correctly identified. This evaluation on real 3D video sequences demonstrates that the temporal similarity...
identifies similar frames and is robust to complex movement and loose clothing. Inter-person similarity across several people each performing a walking motion together with an example similarity curve are shown in Figure 4.8. The similarity measure correctly identifies frames with a similar shape and motion for each person. This illustrates that the temporal similarity measure can also be used to identify similar frames across different people.

4.4 Summary

A comprehensive performance evaluation of shape similarity metrics for 3D video sequences of people has been presented. Existing static shape similarity metrics which give good performance for rigid shape retrieval have been evaluated: shape-distribution [91]; spin-image [58]; shape-histogram [3]; and spherical harmonics [62]. Temporal shape similarity metrics are presented to overcome the ambiguity in independent frame-to-frame comparison. Three approaches are evaluated based on extension of shape-histograms over time: time-filtering of the static shape similarity metric; and shape-flow with local and global frame alignment.

Evaluation of static shape similarity metrics demonstrates that shape-histograms with volume-sampling consistently give the best performance for different actors and motions. However, all static shape similarity metrics are shown to exhibit temporal ambiguities in 3D video for frames with similar shape but different motion directions. Evaluation of temporal shape similarity metrics for a variety of synthetic motions demonstrates that global shape-flow consistently gives the best performance for different people, motions and temporal window size. However, the global shape-flow has an order of magnitude increase in computational cost over time-filtering and local shape-flow. Time-filtered shape histograms are computationally efficient and give marginally lower performance for straight line motions but have significantly reduced performance for non-linear movements (motion with non-linear change of directions). Shape-flow with local alignment achieves comparable performance to global shape-flow, overcoming the limitations of time-filtered static similarity measures for 3D video sequences with non-linear paths, with a computational cost comparable to static shape similarity. However, local frame shape-flow may fail due to errors in alignment at the
central frame whereas global shape-flow is robust.

Evaluation on real 3D video sequences for 9 people demonstrates that time-filtering shape histograms correctly identify frames with similar shape and motion for loose clothing (skirts), complex motions (street-dance) and between different people. Self-similarity also identifies the periodic structure in the motion such as walking and running even for sequences with loose clothing. Performance evaluation on a comprehensive set of real and ground-truth 3D video sequences of people shows that time-filtered shape-histograms are consistent for different people and movements giving a good trade-off between correct similarity and computational cost. However, this similarity metrics only measure geometric similarity and ignore appearance. To be more distinctive and accurate, the following chapter takes both geometric (shape) and appearance (colour) similarity together into account and introduces shape-colour descriptors.
Figure 4.7: Intra-person similarity measure for Real Data. Similarity matrix, curve, example frames for (a) Roxanne Game Character's "Walk"; (b) Roxanne Fashion1's "Walk"; (c) Roxanne Fashion2's "Walk".
Figure 4.7: Intra-person similarity measure for Real Data (continued). Similarity matrix, curve, example frames for (d) Gregor’s “Walk”; (e) Rachel’s “Walk”; (f) Rachel’s “Jog”.
Figure 4.7: Intra-person similarity measure for Real Data (continued). Similarity matrix, curve, example frames for (g) JP's "Pop"; (h) JP's "Lock".

Figure 4.8: Inter-person similarity measure for Real Data. Similarity matrix, curve, example frames for "Walk" across 7 people including Adrian, Gordon, Gregor, Rachel, Jon, Rafael and Tony.
Chapter 5

Shape-Colour Similarity

In this Chapter, we will incorporate photometric similarity (appearance) to the best performer of geometric similarity to provide a more distinctive measure, a volume-sampling spherical Shape-Colour Histogram for the task of matching temporal 3D video sequences of people. The performance of novel Shape-Colour Histograms are compared with both shape-only and other shape-colour descriptors and evaluated using ROC curves.

5.1 Shape-Colour Descriptors

Shape-colour descriptors are a natural extension of 3D shape descriptors to exploit both geometric and photometric information. 3D shape descriptors represent the geometric features of the object which is discriminative enough in many tasks, e.g., to distinguish a chair from a car. However, 3D shape descriptors do not incorporate surface appearance which is an important cue in distinguishing real objects. This limits the object comparison resulting in failure for some simple task. For example, shape-only descriptors cannot distinguish a red ball from a blue ball. Another example is that of two cola cans in different orientation, the task is to match them exactly, Shape Histograms can only match their vertical axis but are unable to resolve the rotation around the vertical axis.

Advances in reconstruction from multiple view video sequences have resulted in highly realistic 3D video sequences of actor performance. Reuse of captured 3D video se-
quences for animation by concatenating captured sequences for different motions requires the identification of transitions points which minimise discontinuities in surface shape and appearance to avoid visual artifacts. High-quality concatenative animation without visual artifacts requires accurate matching of both shape and appearance. For example, concatenating a person walking in direct sunlight with the same motion in shadow or with the sun from a different direction will produce unacceptable visual discontinuities at transitions even if the shape and motion are perfectly matched. In the following sections, the best performer in previous evaluation of Chapter 3, Shape Histograms, is extended to Shape-Colour Histograms taking both geometric and photometric similarity into account. The performance is then compared with three other shape and shape-colour descriptors.

5.1.1 Shape-Colour Histograms

Previous evaluation of 3D shape similarity metrics (Chapter 3) has shown that a volume-sampling spherical Shape Histograms (SHvr) [52] allows identification of 3D video frames with similar surface shape. However, Shape Histograms do not take into account differences in surface appearance which commonly occur in reconstructed 3D video due to non-uniform scene illumination, specular or anisotropic surface reflectance, and non-rigid surface deformation. For Surface Motion Capture (SurfCap) [102], the reconstructed 3D video comprises a 3D surface mesh (shape) together with multiple-view images (colour) at each time frame to preserve the original view-dependent appearance of the captured multiple view video for rendering. Shape-Colour Histograms are introduced to represent both the geometric and photometric features of a 3D object as the joint distribution of its occupancy in colour and space. The 3D object space is partitioned into disjoint cells corresponding to joint spatial and colour bins. Colour information is then localised in each spatial bin by calculating the colour distribution of surface voxels. This gives a 6D histogram description of shape and colour. Figure 5.1 shows an illustration of the Shape-Colour Histograms. Given a 3D video frame $p = < M, I >$, $M = < V, F >$ is a 3D triangle mesh of and $I = \{I_i\}_{N_c}$ a set of multiple-view images, $N_c$ the number of camera views, a volume-sampling spherical Shape-Colour Histogram is constructed as follows:
1. A volumetric representation is constructed by rasterising the surface into a set of voxels that lie inside the 3D object, where the same process is applied as presented in Section 3.1.3, Chapter 3. These occupied voxels' centroid consist a set \( O = o_i, i = 0, ..., N_o - 1 \). \( N_o \) denotes the total number of occupied voxels.

2. Space in Cartesian coordinate system is transformed to a Spherical coordinate system defined by the the centre of mass for the model and vertical axis. For each occupied voxel centroid \( o \in O \), the spherical coordinates \( s = (r, \theta, \phi) \) are calculated the same way presented in Section 3.1.3, Chapter 3. This gives a set of spherical coordinates of occupied voxel centroids \( S = \{s_k\}, k = 0, ..., N_o \). The corresponding colour \( c = (R, G, B) \) for a voxel \( o \) can also be found. We back projecting a point in 3D scene into a camera-view image. Since the calibration is known for each camera, a projection matrix \( P \) is given for all cameras. A 3D scene point, here, \( o = (x, y, z) \) is projecting back to the image \( I \) as follows,

\[
\begin{pmatrix}
    u \\
    v \\
    1
\end{pmatrix}
= \begin{bmatrix}
    P(3 \times 4)
\end{bmatrix}
\begin{pmatrix}
    x \\
    y \\
    z \\
    1
\end{pmatrix}
\tag{5.1}
\]

where \((u, v)\) is the image coordinates and \( I(u, v) \) gives the corresponding RGB colour for \( o \). If the voxel is visible, \( c = I(u, v) \), otherwise, black colour is assigned, \( c = (0, 0, 0) \). Applying this process to each occupied voxels \( o_k \in O \) across all camera views, this gives a set of RGB colour collections \( C = \{c_{ki}\}, k = 0, ..., N_o - 1, i = 0, ..., N_t - 1 \).

3. A 6D spherical histogram \( H(S, C) = [H_{sc}(S, C)]_{N_r \times N_b \times N_s \times N_R \times N_G \times N_B} \) is constructed, accumulating the voxels in the volume representation,

\[
H_{sc}(S, C) = \sum_{k=0}^{N_o-1} \sum_{i=0}^{N_t-1} g(s_k, c_{ki})
\tag{5.2}
\]

\[
g(s_k, c_{lk}) = \begin{cases} 
1 & \text{if } (s_k, c_{lk}) \text{ in } l\text{th bin} \\
0 & \text{otherwise}
\end{cases}
\tag{5.3}
\]
The 1\textsuperscript{st} bin is defined as a subspace,

\begin{equation}
1\textsuperscript{st} \text{bin} := [l_r \cdot \Delta r, l_r \cdot \Delta r + \Delta r) \times [l_\theta \cdot \Delta \theta, l_\theta \cdot \Delta \theta + \Delta \theta) \times [l_\phi \cdot \Delta \phi, l_\phi \cdot \Delta \phi + \Delta \phi) \\
\times [l_R \cdot \Delta R, l_R \cdot \Delta R + \Delta R) \times [l_G \cdot \Delta G, l_G \cdot \Delta G + \Delta G) \times [l_B \cdot \Delta B, l_B \cdot \Delta B + \Delta B)
\end{equation}

(5.4)

where $\Delta r, \Delta \theta, \Delta \phi, \Delta R, \Delta G, \Delta B$ denote the bin size for radius, inclination angle, azimuth angle, colour value for red, green and blue respectively,

\begin{equation}
\Delta\{r, \theta, \phi, R, G, B\} = \frac{\{r, \theta, \phi, R, G, B\}_\text{up} - \{r, \theta, \phi, R, G, B\}_\text{low}}{N_{\{r, \theta, \phi, R, G, B\}}} 
\end{equation}

(5.5)

4. The final descriptor $SCH(M, I)$ is a 6D histogram of the probability normalised by dividing the total number of votes $N_o N_I$,

\begin{equation}
SCH(M, I) = \frac{H(S, C)}{N_o N_I}
\end{equation}

(5.6)
5.1. Shape-Colour Descriptors

5.1.2 Descriptors for Comparison

To compare the performance with other shape and shape-colour descriptors, three previous works are implemented as follows:

- 3D Shape Histogram: This approach is described in Section 3.1.3 which outperforms Shape Distribution, Spin Image and Spherical Harmonics in the task of matching a temporal varying 3D surface sequences of people. Given two 3D video frames $p$ and $q$, we denote the corresponding 3D Shape Histogram descriptors as $SH(p)$ and $SH(q)$ which are 3D histograms.

- LightField [19]: This approach represents a 3D model as a collection of 2D images which are rendered from cameras on a sphere. The main idea is that if two 3D models are similar, they also look similar from all camera views. Accordingly, the similarity between two 3D models can be measured by summing up the similarity from all corresponding images. Here, the image metric is then defined to compute $L_2$ distance between corresponding images. For shape-only LightField descriptor, we use silhouette images and for shape-colour colour images. In our implementation, instead of employing 20 views located on vertices of a regular dodecahedron and consider all possible rotations which is suggested in [19], we use 8 views equally distributed on a circle and the rotation of the camera system is limited to rotating around a vertical axis. Since the quantitative evaluation is performed on a synthetic database, all the models are in a upright orientation, this simplified LightField descriptor will not affect the performance. Given two 3D video frames $p$ and $q$, we denote the corresponding LightField descriptors as $LF(p) = B(I_p)$ and $LF(q) = B(I_q)$ which are sets of multiple-view silhouettes. $B(I_p)$ and $B(I_q)$ compute binary-value images where foreground pixel is set to white and background pixel black. The LightField descriptors considering colour images are denoted as as $LFC(p) = I_p$ and $LFC(q) = I_q$ which are sets of multiple-view colour images.

A temporal filter is finally applied to incorporate temporal similarity.
5.2 Similarity Metrics

The descriptors presented in the previous section can be used to define a similarity measure. Given two 3D video frames \( p \) and \( q \), the shape-colour similarity \( s(p, q) \) is defined as follows: For LightField using silhouettes/colour images, 

\[
s(p, q) = \sum_{t=0}^{N_t-1} \mathcal{E}(I_p, I_q)
\]

where \( \mathcal{E}(\cdot) \) compute the Mean-Square Error (MSE) between images,

\[
\mathcal{E}(I_p(u, v), I_q(u, v)) = \sum_{u=0}^{N_u-1} \sum_{v=0}^{N_v-1} |I_p(u,v) - I_q(u,v)|
\]

where \( I_p(u, v), I_q(u, v) \) denote pixel value at \((u, v)\) in images whose size is \( N_u \times N_v \). \(|\cdot|\) computes the Euclidean distance. Note that \( I_p(u, v), I_q(u, v) \) can be assigned 0 or 1 for silhouettes or RGB colour for colour images. Neither 6D Shape-Colour Histogram descriptor nor 3D Shape Histogram descriptor is rotation invariant, we have to re-define \( s(p, q) \) for 6D Shape-Colour Histogram descriptor to take rotation into account, similar to the definition of similarity metrics for 3D Shape Histogram descriptor in Section 3.2, Chapter 3. For 6D Shape-Colour Histograms,

\[
s(p, q) = \min_{\phi} |SCH(M_p) - SCH(R(M_q, \phi))|
\]

where \( R(M_q, \phi) \) denotes the 3D video frame \( q \) is rotated by \( \phi \) around the vertical axis. Here, we assume human models have an upright direction, since we consider a human pose laying on the ground to be different from a standing pose, even though their shapes are similar since they cannot be concatenated seamlessly.

In practice, we compute the minimal by shifting the histogram with an array \( \phi_n = [0, 1, ..., 359] \), similar to that presented in Section 3.2, Chapter 4: A high-resolution Shape-Colour Histogram \( SCH^*(M_q) \) is constructed and stored; the minimal similarity against different \( \phi \) is computed by shifting \( SCH^*(M_q) \) in dimension \( \phi \) and re-bin it back to \( SCH(M_q) \) for comparison. Since we only consider the rotation about the vertical axis, the bin size of high-resolution Shape Histogram is set to only increase the resolution in dimension \( \phi \),

\[
(\Delta r^*, \Delta \theta^*, \Delta \phi^*, \Delta R^*, \Delta G^*, \Delta B^*) = (\Delta r, \Delta \theta, 1^\circ, \Delta R, \Delta G, \Delta B)
\]
Therefore, we can compute the minimal by shifting the histogram with an array \( \phi_n = [0, 1, \ldots, 359] \),

\[
s(p, q) = \min_{\phi_n} |SCH(M_p) - B(SCH^*(M_q, \phi_n))| \quad (5.11)
\]

where \( B(\cdot) \) denotes rebinning the high-resolution Shape Histogram \( SCH^*(\cdot) \) back to \( SGH(\cdot) \) and \( SCH^*(M_q, \phi_n) \) shifting \( SCH^*(M_q) \) with \( \phi_n \) bins in the dimension of \( \phi \). This evaluates all possible rotation around the vertical axis with 1° resolution.

Finally, we can define a frame-to-frame similarity matrix. Given two 3D video sequences \( P = \{p_i\}_{N_p} \) and \( Q = \{q_i\}_{N_Q} \), the frame-to-frame similarity matrix \( S \) is defined as follows,

\[
S := (s_{ij})_{N_p \times N_Q} \quad (5.12)
\]

\[
s_{ij} = s(p_i, q_j) \quad (5.13)
\]

Previous evaluation of temporal shape descriptors in Section 4.2 showed that efficient classification of surface motion can be achieved by temporal filtering of the frame-to-frame similarity without a significant increase in computational cost. We define a temporal similarity matrix by applying a time filter, the same as previous defined in Section 4.1.1,

\[
s_{ij}^t = \frac{1}{2N_t + 1} \sum_{k=-N_t}^{N_t} s(i+k)(j+k) \quad (5.14)
\]

where a time filter is with window size \( 2N_t + 1 \).

### 5.3 Performance Evaluation

The performance of the shape descriptors and shape-colour descriptors is evaluated against a ground-truth dataset from simulated data. Temporal mesh sequences with rendered images are constructed to mimic 3D video sequences for different motions and the classification of correct and incorrect similarity is assessed using the Receiver-Operator Characteristic (ROC) curves for each technique.

#### 5.3.1 Ground Truth

A simulated dataset is created by using an articulated character model animated from skeletal motion capture data. The model is reconstructed and textured from a real
Chapter 5. Shape-Colour Similarity

Figure 5.2: Simulated dataset cameras and lighting setup: an articulated model in stand pose around by 8 cameras and in front of a spot light.

person with clothing [99]. Since the appearance from the same person does not change, a spot light is created and animated around the model anti-closewise in a circle to change the reflection over a period of 50 frames. Colour images are rendered from eight equally distributed camera views in a circle to mimic the captured multiple-view images of 3D video as shown in Figure 5.2. The following motions are considered: sneak; slow walk; walk circle; fast walk; slow run; run circle; fast run; sprint; rock\&roll dance; and vogue dance, with each motion 150 frames long. The animated model has a single mesh and so the surface correspondence is known at all frames and rigid-body registrations can be performed to align the frames for ground-truth assessment of similarity. This correspondence is only used to compute the “true” surface distance, and is not used in computing the shape or shape-colour similarity measures. Frames from an example synthetic 3D video are shown in Figure 5.3

Given two 3D video frames \( p = < M_p, \{ I_p \} > \) and \( q = < M_q, \{ I_q \} > \), \( i, j = 0, ..., N_l \), we assume they are with the exactly the same 3D shape \( M_p = M_q \), \( p \) without lighting change and \( q \) with periodic lighting change. The sum of image difference in corresponding views \( d_O \) captures purely photometric similarity caused by
5.3. Performance Evaluation

Figure 5.3: Synthetic dataset example frames: images rendered with lighting change from camera 5 for motion Vogue (top), Rock&roll (middle) and Sprint (bottom).

the lighting change,

\[ d_C = \frac{1}{N_1} \sum_{k=0}^{N_1-1} E(I_k^P, I_k^Q) \]  

(5.15)

where \( N_1 \) denotes the total number of views. \( E(\cdot) \) computes the mean-square error between colour images defined in Equation 5.7. For the same model moving under the same changing light, we assume photometric similarity remains the same. The true similarity is then defined as the combination of geometric, photometric and temporal similarity. The geometric similarity is computed as presented in Chapter 3 Section 3.3.1 by finding the average vertex-to-corresponding-vertex distance (Equation 3.37) and the motion similarity by average vertex-to-corresponding-vertex velocity difference (Equation 3.38). Given two 3D video sequences \( P = \{p_i\} \) and \( Q = \{q_j\} \), the position, velocity and appearance difference for a pair of frames are \( d_P(p_i, q_j) \), \( d_V(p_i, q_j) \) and \( d_C \).

A single ground-truth similarity metric is then defined by combination as follows:

\[ s_{ij}^{GT} = (1 - \kappa - \lambda) \cdot d_P(p_i, q_j) + \kappa \cdot d_V(p_i, q_j) + \lambda \cdot d_C \]  

(5.16)

where \( \kappa = \lambda = \frac{1}{3} \) is set to balance geometric, photometric and temporal similarity.
Chapter 5. Shape-Colour Similarity

Figure 5.4: Ground-Truth Generation for Sprint: (a) geometric similarity (Position); (b) temporal similarity (Velocity); (c) photometric similarity (Colour); (d) combination of geometric, photometric and temporal similarity; (d) classification after thresholding.

Ground-truth matching is then defined by taking a single threshold $\tau_{GT}$ for acceptable similarity. If $s_{ij}^{GT} < \tau_{GT}$, similarity classification $c_{ij}^{GT} = 1$ (similar) is defined, denoting that $p_i, q_j$ are similar, otherwise, $c_{ij}^{GT}(p_i, q_j) = 0$ denoting that $p_i, q_j$ are dissimilar. This process is illustrated in Figure 5.4, where geometric, photometric and temporal similarity are combined and then thresholded to provide the ground-truth classification of similarity.

5.3.2 Evaluation of Shape-Colour Descriptors

The ROC performance is evaluated in the same way as that in Chapter 3 Section 3.3.2 and now shown in evaluating the self-similarity for the vogue, rock&roll and sprint in Figure 5.5. The detailed comparison of descriptors is shown in Figure 5.6 for motion sprint. Frames of interest correspond to local minima of the similarity curve and all of them are similar to the query in terms of the 3D shape and motion but Frame 59 and 97
5.4 Similarity Measure on Real Data

In this section, the Shape-Colour Histograms is used to measure shape-colour similarity on real 3D video sequences [102] with a total of 1292 frames. The data comprise a female performer (Roxanne) wearing three different costumes: a game character with shorts and t-shirt (Character1); a long flowing dress (Fashion1); and a shorter tight fitting dress (Fashion2). For each costume 3D video sequences include periodic (walk, run, stagger), motion transitions (walk to run, run to walk, walk to stand, stand to walk) and other motions (hit, twirl). The captured 3D video sequences are unstructured meshes with unknown temporal correspondence and different mesh connectivity and geometry at each time-frame. The Roxanne dataset meshes contain approximately 100k vertices.
Figure 5.6: Comparison for temporal shape-colour and shape-only descriptors. The 1st frame of “Sprint” motion is queried and example frames from camera 5 are shown at the bottom. Threshold is set and draw in dot line to show the difference performance. Points marked on the similarity curves indicate the local minima below given threshold.
and 200k triangles. Multiple-view images are captured from 8 calibrated HD cameras equally spaced in a circle.

The temporal shape-colour histogram is used to evaluate self-similarity and cross-similarity between the 3D video sequences of different motions. The resolution parameters are set as $N_r = 5, N_\theta = 10, N_\tau = 20$ and a temporal window size of $N_t = 2$. Intra-motion self-similarity and inter-motion cross-similarity for each performer together with an example similarity curve are presented in Figures 5.7. The self-similarity clearly identifies the periodic structure of the walking motion even with the highly non-rigid movement of the loose dress and long-hair. Cross-similarity between different motions identifies frames of similar pose and motion. The 3D video frames presented in the graph correspond to minima in the corresponding similarity curve (red points). Evaluation of the reference and matching frames from the 3D video sequence shows that the minima correspond to similar shapes, colour and motion for each 3D video sequence. This demonstrates that the similarity metric correctly recognises sequences of 3D video frames with similar pose, appearance and motion even for loose non-rigid clothing.

5.5 Discussion

There are some issues with Shape-Colour Histograms needed be discussed. First, incorporation of photometric information does not always improve similarity estimation. It is no doubt that two objects with the same geometry but different appearances can be distinguished by introducing Shape-Colour Histograms. However, if there are two objects with similar appearances but different geometries, incorporation of photometric information will reduce the discrimination of geometry. For concatenative animation of the same person, this usually results to a “jump” at transitions since here the discontinuity of geometry is more noticeable than appearance. Therefore, in the following chapters, geometry-only similarity metrics are used for the application of 3D video key-frame extraction and 3D video concatenative animation. Second, when computing the photometric similarity we use the RGB instead of CIE $L^*u^*v^*$ or other colour spaces. For concatenative animation of the same person where in general the appearance is very similar and only the illumination may change, the emphasis of the illumination differences in similarity metrics will help identify optimal transitions where the change
Figure 5.7: Similarity Measure for Real Data using a Shape-Colour Histogram descriptor. Similarity matrix, curve, example frames for (a) Roxanne Game Character's "Walk"; (b) Roxanne Fashion1's "Walk"; (c) Roxanne Fashion2's "Walk"; (d) Similarity Curve for Frame 6 of Roxanne Character1's "Walk"; (e) Self and Cross Similarity Matrix for Roxanne Fashion1's "Walk"; (f) Similarity Curve for Frame 6 of Roxanne Fashion2's "Walk".
of illumination at transitions is imperceptible. Not like CIE $L^*u^*v^*$ or other colour space, the RGB does not intend to be colour constancy and is sensitive to the illumination change. Third, the weighting to balance between geometry and appearance similarity is implied in the way that we counts black pixels within the volume. This is probably not the best way to combine the geometric and photometric similarity. In future, a weighting parameter need to be introduced to explicitly balance them.

5.6 Summary

A novel shape-colour descriptor, shape-colour Histograms, has been introduced as a joint 3D shape, motion and appearance representation. This descriptor allows classification of frames from 3D video sequences reconstructed from multiple view video which are similar in both shape and appearance which provides marginal improvement over shape-only descriptors. The representation extends the volume sampled shape histogram with the addition of colour appearance observed from multiple views. Shape histograms have previously been shown to give the best classification performance for 3D video sequences of people. Orientation invariant similarity between reconstructed frames is achieved by efficient comparison of shape-colour histograms sampled over all orientations. The combination of shape and colour overcomes limitations of previous shape only descriptors which fail to classify 3D shapes with variations in surface appearance. Changes in surface appearance commonly occur in multiple view reconstruction due to non-uniform scene illumination and non-rigid surface deformation. A comparative evaluation of the novel shape-colour histogram against ground-truth synthetic 3D video sequences for ten different motions demonstrates consistent improvements in classification performance against previous shape only and shape-colour descriptors. Shape-colour histograms are applied to the problem of identifying similar frames in real 3D video sequences of actor performance with loose clothing and hair. The descriptor accurately identifies frames with similar surface shape and appearance allowing transitions between different sequences.

In the following chapter, self-similarity will be used to extract key-frames for 3D videos. These key-frames help the user browse and select key-frames to control the concatenative animation production from captured 3D video.
Chapter 6

3D Video Key-Frame Extraction

There are two motivations to develop an automatic 3D video key-frame extraction method: first, automatically extracted key-frames can be used as a means of control over the production of animation, users can simply select from key-frame summarization and avoid wasting time to browse every frame in the 3D video database; second, automatically extracted key-frames can be further used for developing a 3D video compression algorithm as a mimic to 2D video compression [109]. In the field of 2D video compression, only key-frames and changes that occur from them to successive frames are stored which greatly reduce the amount of information that must be stored.

The key-frame extraction strategy is presented based on the shape and motion similarity metrics introduced in Chapter 3-5, either shape-only or shape-colour ones. The basic idea is to identify a set of key-frames which can “best” represent the original motion, in other words, each key-frame can “best” represent its neighbour frames. Selection of frames should be a trade-off between compactness (rate) and faithfulness (distortion). Therefore, we set a trade-off between rate and distortion and then optimise the selection of key-frames by maximising each key-frame’s neighbourhood. The following sections describe how we convert this key-frame optimisation problem into a graph optimisation problem and solve it as a shortest path.
6.1 Background

In earlier work on 2D video, key frames were selected by sampling video frames randomly or uniformly at certain time intervals [107]. This approach is simple and fast but neglects the video content. Therefore the approach may miss representative frames and include redundant frames. To address this problem, shot-based key-frame extraction algorithms have been proposed [123]. A video is first segmented into shots and then key-frames are extracted for each shot independently.

A popular method for key-frame selection is clustering. Similar frames are clustered and a representative frame from each cluster is selected as a key-frame. Campbell et al. [15] used phase-space (2D projections of joint positions and velocities) for recognising atomic ballet moves from motion capture data where temporal correspondence is known and the system learns clusters from training data. Zhuang et al. [124] proposed an unsupervised clustering based on colour histograms to segment video into shots. For each shot the frame closest to the cluster centre is selected as a key-frame. Lagendijk et al. [65] developed a similar approach clustering the video sequence and for each cluster the key frame which minimises the visual redundancy is selected. Loy et al. [74] applied clustering to sports video based on 2D shape context similarity, key frames are chosen as the most central frame in each cluster (the frame with the minimum average within cluster distance is the key frame). Liu et al. [70] store the extracted cluster key frames in an efficient motion index tree, to improve retrieval time of 3D motions with different speeds. Ratakonda et al. [94] proposed a hierarchical video summarization using a pair-wise K-means algorithm. Doulamis et al. [27] adopted a fuzzy classifier to cluster all features extracted through a recursive shortest spanning tree algorithm to predetermined classes. And a genetic algorithm is adopted to extract key-frames by minimising a cross-correlation criterion. Kim and Hwang [63] also present an object-based video abstraction through Mean Shift Clustering.

Grouping similar frames and selecting representative frame also exploits self-similarity. BenAbdelkader et al. [10] regard gait motion as repeated blocks in a self-similarity matrix, constructed from distances between sequences of simultaneously scaled silhouettes. Cooper et al. [21] decompose a self-similarity matrix from distances between DCT coefficients to generate video summaries. Vermaak et al. [113] maximise the
dissimilarity between consecutive key frames, and favour frames with high entropy.

Another popular key frame extraction is curve simplification. Ramer et al. [93] constructs a polygonal approximation to a curve, by repeatedly splitting the line at the point with the maximal distance from the curve. DeMenthon et al. [25] applied curve simplification to video DCT coefficients, while Lim et al. applied it to 3D motion capture data [69]. They reported that a reasonable approximation can be made with about a fifth of the frames. Finally, Xiao et al. [116] provide a keyframe extraction method based on a novel layered curve simplification algorithm for motion capture data, where features such as bone angles are tracked.

Li et al. [68] extract key frames from 2D video by a rate-distortion optimisation. The optimal algorithm is based on dynamic programming and practical constraints such as the maximum rate or distortion should be pre-defined by a user. Xu et al. [118] also consider rate-distortion trade off for a 3D video summarization, however, shots detection must be performed before key frame extraction as a pre-processing and a user defined control parameter is required. The basic idea is to segment 3D video into shots based on the motion (the amount of changes) of 3D object, similar idea to the proposed key-frame extration. A distance (similarity) is calculated between successive frames to reveal motion after extracting the feature vectors for each frame. In [117], the feature vectors are based on three histograms of all the mesh vertices in spherical coordinate system, where the vertex positions are transformed to the spherical coordinate system, similar to presented 3D Shape Histogram descriptors in Chapter 3 Section 3.1.3. The 3D video is then segmented into shots with a ground-truth decision of eight independent assessors. The proposed key-frame extraction method is fully automatic and does not require a pre-defined shots detection.

Many key frame extraction methods mentioned above are based on a two-step approach: an initial shots detection step, followed by independent key-frame extraction for each shot. The quality of key-frame extraction depends on the quality of shot detection. For clustering methods, the chosen centre of the cluster may not work, since motion is better represented by local extreme points which tend to be off centre [6]. Curve simplification requires motion analysis to establish temporal correspondence, for example, tracking features, which is easy to do with motion capture data but difficult with 3D video data.
Establishing temporal correspondence is also computationally expensive as a basis for summarization.

6.2 Similarity-based Key-frame Extraction

We use the self-similarity measure between all frames of a 3D video sequence to extract key frames directly without an intermediate shot detection. The summarization of the 3D video will be the set of key frames extracted. In the following sections, we first define the Rate, the Distortion, and their weighted sum, the Conciseness Cost. We construct a graph according to the Conciseness Cost, and the key frame extraction is converted to a shortest path problem. Finally, we automatically determine an optimal selection of key frames in the sense of the trade-off between the Rate and the Distortion.

6.2.1 Rate and Distortion

We propose a definition of rate and distortion for 3D video summarization, similar to the definitions in [118]. Generally, the rate should be the entropy of key frames and the distortion should be the information loss between the key frames and the original sequence. The rate $R$ is defined as the number of key frames $N_k$,

$$R^K = N_k$$  \hspace{1cm} (6.1)

The distortion $D$ for the entire sequence is defined as the sum of distortion for each key frame representing its adjacent frames. Given a set of indices of key frames $K = \{k_j\}_{j=1}^{N_k}$ to represent a 3D video sequence $P = \{p_i\}_{i=1}^{N_p}$, and assuming $p_{kj}$ represents adjacent frames $\{p_{kj-N_{kj}},...,p_{kj+N_{kj}}\}$ the distortion for key-frame $p_{kj}$ is

$$D^K_{kj} = \sum_{l=-N_{kj}}^{N_{kj}} s(p_{kj}, p_{kj+l}) = \sum_{l=-N_{kj}}^{N_{kj}} s_{kj,kj+l}$$  \hspace{1cm} (6.2)

where self-similarity matrix for motion $P$ is $S = [s_{ij}]_{i=1...N_p}$. The total distortion $D$ for the whole sequence is

$$D^K = \sum_{j=1}^{N_k} D^K_{kj}$$  \hspace{1cm} (6.3)
The quality of a summarization depends on two costs: the representative cost (Rate) and the accuracy cost (Distortion). We define the Conciseness Cost $C$ as a weighted sum of the Rate and Distortion,

$$C^K = \beta \cdot R^K + (1 - \beta) \cdot D^K$$ (6.4)

where $\beta$ is the parameter to weight the Rate and Distortion. A good summarization prefers a small Conciseness Cost. Given $\beta$, the optimal selection of key frames $K_{\beta}^{opt}$ is the one which minimises the Conciseness Cost,

$$K_{\beta}^{opt} = \arg\min_{K \leq \{1, \ldots, N_p\}} \{C^K\}$$ (6.5)

### 6.2.2 Conciseness Cost Matrix

Before we construct the graph, we pre-compute a Conciseness Cost Matrix,

$$C := (c_{ij})_{N_p \times \lceil \frac{N_p+1}{2} \rceil}$$ (6.6)

$$c_{ij} = (1 - \beta) \cdot d_{ij} + \beta \cdot 1$$ (6.7)

where $\beta$ is the parameter to weight the Rate and Distortion and $d_{ij}$ is the entry of a Distortion Cost Matrix $D = (d_{ij})_{N_p \times \lceil \frac{N_p+1}{2} \rceil}$ which can be derived from the Self-Similarity Matrix $S$

$$d_{ij} = \sum_{n=-j}^{j} s_{i+i-n}$$ (6.8)

Given $\beta = 0.5$, Conciseness Cost Matrices for Roxanne's Hit and JP's Lock are shown in Figure 6.1. Each element of the matrix $C$ represents the cost of selecting each frame in a motion sequence as a key-frame that spans a specific range of frames. The task is then to select the optimal number of key-frames that span the motion sequence and minimise the total cost.

### 6.2.3 Graph Construction

A graph can be constructed from the Conciseness Cost Matrix. The optimisation for the location of each key frame and the number of adjacent frames it represents converts to a shortest path problem in the graph. Each entry of the Conciseness Matrix is a node in the graph. The connections and edge distance is defined as follows, edges are
Conciseness Cost Matrix: Roxanne-Hit

Conciseness Cost Matrix: JP-Lock

Chapter 6. 3D Video Key-Frame Extraction

Figure 6.1: Conciseness Matrix for Roxanne’s Hit (left) and JP’s Lock (right), $\beta = 0.5$.  

formed between each frame node and all nodes which could represent it. Example for a 5 frames sequence is shown in Figure 6.2. Let us denote $Node(i',j')$ as the node corresponding to row $i'$ and column $j'$ in the conciseness matrix then:

1. Each entry $c_{i',j'} \in C$ corresponds to a node $Node_p(i',j')$ in the graph. $Node_p(i',j')$ denotes that frame $i'$ is a key-frame candidate representing $j'$ frames in its past and future. If $j' = 0$ then frame $i'$ represents itself only.

2. The edge distance from $Node_p(i',j')$ to $Node_q(m',n')$ is defined as $Edge(p,q) = c_{m',n'}$. $Edge(p,q)$ denotes the Conciseness Cost introduced by including $Node_q(m',n')$ in the path.

3. Key frames must be in time ascending order. If $i' \geq m'$ then $Edge(p,q) = \infty$.

4. All frames in the sequence must be represented by one key frame. If $i' < m'$ and $i' + j' < m' - n'$ then $Edge(p,q) = \infty$.

5. The source node $Node_{source}$ is added and connected with all nodes which can represent the first frame. If $i' - j' = 1$ then $Edge(source,p) = c_{i',j'}$ else $Edge(source,p) = \infty$.

6. The sink node $Node_{sink}$ is added and connected with Nodes which can represent the last frame. If $m' + n' = N_z$ then $Edge(q,sink) = 0$ else $Edge(q,sink) = \infty$. 
6.2. Similarity-based Key-frame Extraction

Figure 6.2: Example Graph for a sequence of 5 frames. Obviously, there are only 4 possible graph paths from source to sink: $1 \rightarrow 2 \rightarrow 4 \rightarrow 7 \rightarrow 9$ (corresponding key-frames are 1,2,3,4,5, each key represents itself), $3 \rightarrow 7 \rightarrow 9$ (corresponding key-frames are 2,4,5, key-frame 2 represents frame 1,2,3), $1 \rightarrow 5 \rightarrow 9$ (corresponding key-frames are 1,3,5, key-frame 3 represents frame 2,3,4), $1 \rightarrow 2 \rightarrow 8$ (corresponding key-frames are 1,2,4, key-frame 4 represents frame 3,4,5), 6 (corresponding key-frame is 3 which represents frame 1,2,3,4,5).

6.2.4 Shortest Path

Given $\beta$ an optimal solution can be found via search for the shortest path in the constructed graph. Here, we apply Dijkstra algorithm [26] to find the shortest path from $Node_{source}$ to $Node_{sink}$ which contains all the key frames and covers the entire sequence. This gives a global optimal key frames extraction for a given $\beta$. Figure 6.3 shows the shortest path on the Self-Similarity Matrix (rotated by 45°) for Roxanne’s Hit and JP’s Lock respectively. We can see that for slow changes where larger regions have a low cost then only one key-frame is selected, where as for fast changes more key-frames are selected. Corresponding sequence and key frames are shown in Figure
6.3 3D Video Key-Frame Extraction

Figure 6.3: Shortest Path shown as white line on the Self-Similarity Matrix for Roxanne's Hit (left) and JP's Lock (right), $\beta = 0.5$.

6.4 (a) and (c).

6.2.5 Optimal Point on the R-D curve

The Rate-Distortion curve is guaranteed to decrease monotonically. As the rate of key-frames increases the resulting distortion will decrease. We therefore automatically define the operating point on the rate-distortion curve where an increase in key-frame rate has a relatively reduced effect on decreasing distortion. The units of our rate and distortion measure are equal and the optimal point can be defined simply where

$$\frac{dD}{dR} = -1$$

This corresponds simply to $\beta = 0.5$, where rate and distortion are treated equally in defining conciseness.

6.3 Experimental Results and Evaluation

In this section, we present experimental results of our key-frame extraction and compare them with baseline key-frame extraction used in [118]. The key frames for a female subject (Roxanne) performing nine short sequences of motion including Hit (8 key-frames extracted from 76 frames in total denoted by 8/76), Stagger (9/81), Walk to
6.4 Discussion

Stand (6/56), Stand (2/50), Stand to Walk (5/31), Walk to Jog (5/27), Jog (4/26), Jog to Walk (4/28) and a male subject (JP) performing three long sequences including Lock (22/250), Lock to Pop (26/250) and Pop (20/250) are shown in Figure 6.4 and 6.5. Key-frames are selected according to our R-D optimal criteria. The baseline key-frame extraction [118] is set at the same Rate. For short sequences, which can be regarded as shots after shot detection, our key-frames extraction and baseline method both work well giving qualitatively reasonable summarization, but our method gives less redundant more representative key-frames, shown in Figure 6.4(a)(b). For long sequences, which can be regarded as 3D video without shot detection, our method outperforms baseline method giving less redundant key-frames and keeping more motion information, shown in Figure 6.4(c). For all tested sequences, the proposed optimisation selects key-frames according to the movement dynamics resulting in increased spacing of key-frames for slow movement and identification of suitable key-frames to represent visually significant changes in posture.

The performance of our key-frame selection approach is compared to the method proposed in [118]. Rate-distortion characteristics for the two approaches applied to the twelve different female (Roxanne) and male (JP) 3D video sequences are presented in Figure 6.6. These rate-distortion characteristics are provided by computing the optimal solutions for $\beta$ in the range of $[0, 1]$. The optimal key-frame solution is marked as a dot on the R-D curve. Comparison of the R-D characteristics for our approach with the that of [118] demonstrates improved performance for all 3D video sequences with a lower R-D curve for the key-frames generated by our approach. There are significant large improvements with our method, expect the case of Roxanne’s Stand which is almost still, as the method in [118] does not use the 3D video dynamics resulting in redundant key-frames.

6.4 Discussion

There are some issues with Self-Similarity based Key-Frame Extraction needed be further clarified. First, a key frame here is defined as a representative frame of its neighbourhood which is not necessarily a key pose/event (e.g. a heel contact frame) and the purpose is to summarise a 3D video sequence with an optimal balance between
(a) Original 3D video sequence (top smalls), Proposed Key-frame extraction (middle row), compared with Baseline (bottom row) [118] for Roxanne’s Hit.

(b) Proposed Key-frame extraction (top row), compared with Baseline (bottom row) [118] for Roxanne’s Stagger.

(c) Proposed Key-frame extraction (top two rows), compared with Baseline (bottom two rows) [118] for JP’s Lock

Figure 6.4: Proposed key-frame extraction vs. Baseline for Roxanne and JP 3D video.
6.5 Summary

Figure 6.5: More key-frame extraction for Roxanne and JP 3D video: Walk to Stand, Stand, Stand to Walk (top); Walk, Walk to Jog, Jog (middle); Lock to Pop (bottom).

conciseness (rate) and faithfulness (distortion). Second, the adoption of this approach over clustering approaches is because that the clustering usually requires a number of clusters or a bandwidth and how to choose that is arbitrary, however, the proposed approach only requires a weighting parameter which explicitly balance between rate and distortion. Third, the computational complexity of the shortest path algorithm (Dijkstra) is $O(n^2)$ ($n$ denotes the total number of frames in the original sequence) which limits the scalability to a longer sequence. In future, other shortest path algorithms with less computational complexity will be investigated.

6.5 Summary

In this chapter, we propose an automatic key-frame extraction method for 3D video summarization. Volume-sampling spherical Shape Histograms are adopted to compute the self-similarity of 3D video sequences. A graph is constructed from the self-similarity and the shortest path is found as the set of key frames. The global optimisation process provides a concise representation. The optimal key-frame summarization is automatically evaluated to balance the representation size (rate) against the representation accuracy (distortion). Experimental results show the summarization is compact and faithful to the original 3D video. Our method has been compared with a baseline method, key-frame extraction for a shot, used in [118]. The results of tested short sequences demonstrate our method gives improved key-frame selection for a variety of 3D video sequences of human movement with different dynamics. Sequences without
shot detection demonstrate our method significantly outperforms the baseline. The approach automatically produces key-frame summarization for 3D video of human movement without requiring any manual parameter adjustment, shot detection or temporal correspondence. The automatic key-frame extraction can be incorporated into the animation system to make user selection of key-frames easier. Similarity metrics (Chapter 3–5) and key-frames extraction (this Chapter) form the basis of the animation from 3D video introduced in the next chapter.
Figure 6.6: Rate-Distortion Curve
Chapter 7

Surface Motion Graphs

The framework for concatenative synthesis from a database of 3D video sequences of human motion comprises two stages: pre-processing the database of the 3D video sequences to construct a *surface motion graph*; and motion synthesis by optimising the graph-path to satisfy user-defined constraints and minimise the transition cost between 3D video segments. The surface motion graph for a set of 3D videos represents the possible transitions between 3D video sequences and self-transitions within a sequence, which is analogous to motion graphs [64] for skeletal motion capture sequences. Similar to Motion graphs, *move trees* [76] are widely used in computer games for animation. Move trees are created manually by first carefully collecting short motion clips and then linking and blending them as a graph which is mainly by hand using interactive tools. Motion Graphs are constructed automatically given skeletal motion capture database which is more desirable. [100, 101, 1, 2, 112, 114, 120]. Current approaches are computationally expensive and require manual intervention to obtain correct correspondences for complex non-rigid surface motion. In this framework, the user can select key-frames from a database of 3D video and set a target traverse distance $dv$ and time $tv$, our system will then automatically generate the animation that can best satisfy the user-defined constraints. Figure 7.1 shows an example of user-defined constraints.
7.1 Building a Surface Motion Graph

The approach to identify and produce transitions between 3D video sequences for constructing surface motion graphs is presented. Transition points between 3D video sequences are identified without temporal correspondence using a volumetric temporal similarity metric (Chapter 3—5). Pre-processing of the 3D video database identifies the optimal location and temporal window length for transitions to maximise the shape similarity between sequences allowing transitions without unnatural intermediate motion. The set of 3D videos is structured into a surface motion graph which represents all possible inter and intra sequence transitions. This representation is subsequently used for motion synthesis according to user-defined constraints.

7.1.1 Transitions

A surface motion transition from 3D video sequence $P$ to $Q$ is defined as an overlap sub-sequence of frames. If $p_m$ and $q_n$ is the first frame overlapped from $P$ and $Q$ respectively and overlap length is $N_T$, the smooth transition $T = \{t_k\}, k = 0, \ldots, N_T - 1$ could be generated by a blending,

$$t_k = \mu(k) \cdot p_{m+k} + (1 - \mu(k)) \cdot q_{n+k}$$

(7.1)

where $t_k$ denote the blended frame of $p_{m+k}$ and $q_{n+k}$ according to weighting function $\mu(k)$. This is illustrated in Figure 7.2. We consider two simple functions $\lambda(k)$ and $\iota(k)$.
7.1. Building a Surface Motion Graph

Figure 7.2: The illustration of a transition concatenating two sequences. Each node denotes a frame, blue original frames, red transition frames.

for the weight function $\mu(k)$,

$$\lambda(k) = 1 - \frac{k}{N_T - 1}$$  \hspace{1cm} (7.2)

$$\nu(k) = \begin{cases} 1 & \text{if } k < \frac{N_T - 1}{2} \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (7.3)

If $\mu(k) = \lambda(k)$, Equation 7.1 becomes a linear blending; if $\mu(k) = \nu(k)$, Equation 7.1 concatenates two sequences by switching at their central frames. Figure 7.2 shows a transition concatenating two sequences and Figure 7.3(a)(b) two weighting functions. Since the process of linear blending requires surface correspondence which is unknown in this work, weighting function Equation 7.2 is only used as a guide to identify transitions. The concatenation is then performed as a switch from $P$ to $Q$ at the centre of the overlapped window according to weighting function Equation 7.3. Subsequent sections apply an adaptive time-filtered Shape Histograms (SHvrT) as a similarity metric to identify transitions and the adaptive temporal filtering is performed according to weighting defined in Equation 7.2.

7.1.1.1 Static Similarity Measure

To identify optimal transitions between 3D video sequences without temporal correspondence we use a time-filtered volumetric shape histogram to define a similarity metric over a temporal window. The time-filtered volumetric shape histogram has pre-
Chapter 7. Surface Motion Graphs

![Weighting functions](image)

(a) Linear weighting function  (b) Switch weighting function  (c) Filter weighting function

Figure 7.3: Weighting functions. (a) weighting for linear blending transitions; (b) weighting for switching at central frames; (c) weighting for temporal filter.

Previously been shown to give good performance for human motion recognition on 3D video sequences of people in Chapter 3. Given a 3D video database \( \mathbb{P} = \{P_k\} \), for a pair of 3D video sequences \( P_i \) and \( P_j \), we first calculate static similarity matrix \( S_{ij} \), adaptive temporal filtering is then exploited to consider temporal similarity and locate a transition \( T_{ij} \). Following sections will explain this process.

7.1.1.2 Adaptive Temporal Filtering

A smooth transition \( T \) from \( P \in \mathbb{P} \) to \( Q \in \mathbb{P} \) can be obtained by performing a linear blending, the quality is determined by the distortion to the original sequences. We define a measure of distortion \( d(t_k, p_{m+k}) \) between a transition frame \( t_k \) and original frame \( p_{m+k} \), similarly, \( d(t_k, q_{n+k}) \) between \( t_k \) and \( q_{n+k} \). Since either \( p_{m+k} \) or \( q_{n+k} \) is from the original sequence, we define the distortion of transition frame \( t_k \) as the minimal distortion of them, i.e.

\[
d(t_k) = \min\{d(t_k, p_{m+k}), d(t_k, q_{n+k})\}.
\] (7.4)

Therefore, the total distortion for a transition \( T \) can be computed as a sum of the distortion of all transition frames,

\[
d(T) = \sum_{k=0}^{N_T-1} d(t_k) = \sum_{k=0}^{N_T-1} \min\{d(t_k, p_{m+k}), d(t_k, q_{n+k})\}.
\] (7.5)

An optimal transition \( T^{opt} \) should minimise the total distortion,

\[
T^{opt} = \arg \min_T d(T)
\] (7.6)
However, without surface correspondences we cannot produce such a transition, \( d(t_k, p_{m+k}) \) or \( d(t_k, q_{n+k}) \) cannot be computed. Alternatively, since the shape similarity between \( p_{m+k} \) and \( q_{n+k} \) is known as \( s_{(m+k)(n+k)} \), we can then approximate the distortion using the weighted similarity. The weight is set according to blending Equation 7.1 with linear weighting Equation 7.2. Therefore, two approximations are computed as follows,

\[
\begin{align*}
    d(t_k, p_{m+k}) & \approx \lambda(k) \cdot s_{(m+k)(n+k)} \\
    d(t_k, q_{n+k}) & \approx (1 - \lambda(k)) \cdot s_{(m+k)(n+k)}
\end{align*}
\]

The distortion for a transition frame \( t_k \) can be approximated as follows,

\[
d(T) \approx \sum_{k=0}^{N_T-1} \min\{\lambda(k) \cdot s_{(m+k)(n+k)}, (1 - \lambda(k)) \cdot s_{(m+k)(n+k)}\}
\]

\[
= \sum_{k=0}^{N_T-1} \min\{\lambda(k), (1 - \lambda(k))\} \cdot s_{(m+k)(n+k)}
\]

\[
= \sum_{k=0}^{N_T-1} \chi(k) \cdot s_{(m+k)(n+k)}
\]

\[\chi(k) = \min(\lambda(k), 1 - \lambda(k)) = \begin{cases} 
    \frac{k}{N_T-1} & \text{if } k < \frac{N_T-1}{2} \\
    1 - \frac{k}{N_T-1} & \text{otherwise}
\end{cases}\]

where \( s(T) \) denotes a temporal filtered similarity with a weighting function \( \chi(k) \) which is shown in Figure 7.3 (c). Accordingly, the optimisation of a transition defined in Equation 7.6 becomes,

\[\arg \min_T s(T)\]

Since \( T \) is determined by a triple set of the starting point of transition and the length of transition \( \{m, n, N_T\} \), the optimisation is then performed by testing all possible triple sets to minimise the filtered similarity,

\[
T^{opt} = \{m^{opt}, n^{opt}, N_T^{opt}\} = \arg \min_{m,n,N_T} \sum_{k=0}^{N_T-1} \chi(k) \cdot s_{(m+k)(n+k)}
\]

where \( 0 \leq m < N_P, 0 \leq n < N_Q \) and \( 1 \leq N_T < \min(N_P, N_Q) \). This process is called an adaptive temporal filtering.
7.1.2 Graph Construction

A surface motion graph is automatically constructed as a two-level directed graph. At the higher-level, each node represents a motion and each edge a transition with a direction. In the lower-level, each node represents a frame and each edge a sequence of frames connecting them. Figure 7.4 shows an example of a surface motion graph for four motions. Given a 3D video database $\mathcal{P} = \{P_k\}$, $k = 0, ..., N_p - 1$, the Surface Motion Graph is constructed as follows:

1. Initialisation: Insert all 3D video sequences as edges and their terminal frames as nodes to create a disconnected graph (Figure 7.4 upper-right).

2. Identify transitions: A transition from $i^{th}$ motion to $j^{th}$ motion is identified by evaluating $T_{ij}^{opt} = \{m_{ij}^{opt}, n_{ij}^{opt}, N_{T_{ij}}^{opt}\}$ according to Equation 7.12. This procedure is performed across all pairs of motions.

3. Insert transitions: At the higher-level, if there is a transition from $i^{th}$ motion to $j^{th}$ motion, automatically insert an edge from $P_i$ to $P_j$. The user can then modify this higher-level graph by eliminating unwanted edges before generate the lower-level graph. At the lower-level, terminal frames of transitions, if they are not in the graph, are then inserted as nodes, breaking existing edges into smaller ones. In-between-terminal frames of transitions are inserted as edges to join either different sequences or different parts of the same sequence.

7.2 Motion Synthesis

Motion synthesis is performed by optimising over all possible paths on the lower-level surface motion graph according to user-defined hard and soft constraints. The hard constraints are key-frames selected by the user which define the desired movement and must be satisfied. Initially, the user is allowed to select a start key-frame and an end key-frame from a given 3D video database. Note that additional key-frames can also be added to refine the synthesised motion. The soft constraints are user-specified target traverse distance $d_V$ and target traverse time $t_V$. These soft constraints are formulated together as a cost function that is described in the following sections. In practice, hard
Figure 7.4: An example of Surface Motion Graph. Double circles denote start and end key-frames. The higher-level graph represents the 3D video sequences and transitions, the lower-level graph represents all possible motion graphs between a particular start and end keyframe.
constraints are first exactly satisfied by inserting key-frames into the lower-level graph as nodes between which soft constraints are then best satisfied by minimising the cost function.

7.2.1 Cost Function

Given a path $F$ which satisfies the key-frame constraints, the cost $C(F)$ through the surface motion graph is defined as a combination of the total transition cost $C_s(F)$, distance cost $C_d(F)$ and time cost $C_t(F)$,

$$C(F) = C_s(F) + w_d \cdot C_d(F) + w_t \cdot C_t(F)$$  \hspace{1cm} (7.13)

where $w_d$ and $w_t$ are weights for distance and time constraints respectively. $w_d = 1/0.3$ and $w_t = 1/10$ are set to equate the penalty for an error of 30 centimetres in distance with an error of 10 frames in time [4] with a relative weight of 1.0 to emphasise the smoothness cost. Individual cost terms for smoothness, distance and time are defined as follows:

**Total Smoothness Cost** $C_s(F)$ for a path $F$ is defined as the sum of dissimilarity for all transitions between concatenated 3D video segments which is computed in the higher-level graph. If we denote the index for concatenated 3D video segments as $\{f_i\}$, $i = 0...N_F - 1$, the total smoothness cost $C_s(F)$ is computed as follows,

$$C_s(F) = \sum_{i=0}^{N_F-2} s(T_{f_i,f_{i+1}})$$  \hspace{1cm} (7.14)

where $N_F$ denotes the total number of transitions on path $F$ and $s(T_{f_i,f_{i+1}})$ the filtered dissimilarity for transition from motion sequence $I_{f_i}$ to motion sequence $I_{f_{i+1}}$.

**Distance Cost** $C_d(F)$ for a path $F$ with $n_F$ frames is computed in the lower-level graph as the absolute difference between the user-specified target distance $d_V$ and the total travelled distance $d(F)$, given the 3D video frames on the path of $F$ is $\{J_j\}$, $j = 0...n_F - 1$

$$C_d(F) = |f_d(F) - d_V|$$  \hspace{1cm} (7.15)

$$f_d(F) = \sum_{j=0}^{N_F-2} |c(J_{j+1}) - c(J_j)|$$  \hspace{1cm} (7.16)
where function \( c(J_{j+1}) \) and \( c(J_j) \) denotes the projection of the centroid of the mesh at frame \( J_{j+1} \) and \( J_j \) onto the ground respectively along the vertical axis.

**Time Cost** \( C_t(F) \) for a path \( F \) with \( N_f \) frames is evaluated as the absolute difference between the user-specified target time \( t_V \) and the total travelled time \( t(F) \),

\[
C_t(F) = |f_t(F) - t_V| \tag{7.17}
\]

\[
f_t(F) = n_F \times \Delta t \tag{7.18}
\]

Here, the frame rate \( \Delta t \) of captured 3D video sequences is \( \Delta t = \frac{1}{25} \).

### 7.2.2 Path Optimisation

In this section, we present an efficient approach to search for the optimal path that best satisfies the user defined soft constraints. The optimal path \( F^{opt} \) is found to minimise the combined cost \( C(F) \) defined by Equation 7.13,

\[
F^{opt} = \arg\min_F \{C(F)\} \tag{7.19}
\]

Enumerating all possible paths and evaluating the combined cost will give the global optimal. But when cycles appear in the graph, the number of paths may become infinite. To avoid this, instead of enumerating all paths, we enumerate all walks (paths without any loop) which satisfy key-frame constraints and all loops attached to each walk. The global optimal path must be a composition of a walk and attached loops. Since the number of walks are finite, the optimisation can be performed in two steps. First, for each walk, we optimise the choice and repetitions of attached loops; second, we compare all walks with their own optimal loops configuration and find the optimal walk.

To optimise over the number of repetitions of the graph loops for a particular walk, we first denote the walk \( l_0 \) with attached loops \( \{l_1, \ldots, l_{N_L}\} \) as \( L = [l_i]_{N_L} \) and corresponding number of repetitions \( n = [n_i]_{N_L} \), \( n_0 = 1 \). A path \( F \) can then be represented as a composition of a walk and attached loops and denoted as

\[
F = n \cdot L \tag{7.20}
\]
The optimisation then becomes,

\[ F_{\text{opt}} = n_{\text{opt}} \cdot L_{\text{opt}} = \arg\min_{n \cdot L} \{ C(n \cdot L) \} \]  \hspace{1cm} (7.21)

Essentially, a walk or a loop is a graph path which represents a sequence of frames. The computation of smoothness, distance and time costs are of no difference. Therefore, cost functions \( f_d(l), f_t(l), f_s(l) \) defined in Section 7.2.1 can be used to compute smoothness, distance and time cost for \( L \). The smoothness, distance and time cost for \( n \cdot L \) are computed as follows,

\[ C_s(n \cdot L) = |n \cdot f_d(L) - d_v| \]  
\[ C_d(n \cdot L) = |n \cdot f_t(L) - t_v| \]  
\[ C_t(n \cdot L) = n \cdot f_s(L) \]  \hspace{1cm} (7.22) \hspace{1cm} (7.23) \hspace{1cm} (7.24)

and the combined cost defined in Equation 7.13 becomes

\[ C(n \cdot L) = C_s(n \cdot L) + w_d \cdot C_d(n \cdot L) + w_t \cdot C_t(n \cdot L) \]  \hspace{1cm} (7.25)

Once the surface motion graph is constructed and key-frames are selected, \( L \) is determined, the goal becomes to optimise \( n \) by minimising \( C(n \cdot L) \) defined in Equation 7.25 and Equation 7.24.

In the following paragraphs a two-step optimisation is performed. Let \( N_k \) denote the number of walks, for \( k \)th walk \( L_{k,0} \), \( L_k \) is determined and the objective is to find corresponding optimal repetitions of loops \( n_{k,0}^{\text{opt}} \) according to Equation 7.21,

\[ n_{k,0}^{\text{opt}} = \arg\min_{n_k} \{ C(n_k \cdot L_k) \} \]  \hspace{1cm} (7.26)

Let \( k_{\text{opt}} \) denote the index of the global optimal walk, we enumerate all walks and compare their optimal cost and find the minimal

\[ k^{\text{opt}} = \arg\min_{k=1, \ldots, N_k} \{ C(n_{k,0}^{\text{opt}} \cdot L_k) \} \]  \hspace{1cm} (7.27)

Finally, the optimal path \( F_{\text{opt}} \) is a composition of the optimal walk and loops \( L_{\text{opt}} = L_{k^{\text{opt}}} \) together with optimal repetitions for each loop \( n_{\text{opt}} = n_{k^{\text{opt}}}^{\text{opt}} \),

\[ F_{\text{opt}} = n_{k^{\text{opt}}}^{\text{opt}} \cdot L_{k^{\text{opt}}} \]  \hspace{1cm} (7.28)
7.2. Motion Synthesis

Figure 7.5: Examples of Graph Path Decomposition that consider nested sub-loops.

The decomposition of an arbitrary path to a walk and attached loops will be described in Section 7.2.2.1 and the composition of a walk and attached loops back to a path in Section 7.2.2.3. The optimisation of repetitions for a given walk and loops according to Equation 7.26 are solved as Integer Linear Programming (ILP) problems in Section 7.2.2.2.

7.2.2.1 Graph Path Decomposition

Depth-First Search (DFS) [108] is exploited to decompose all possible paths (constrained by key-frames) to walks and loops. Given an adjacency matrix \( A \) for the graph, source and sink (start and end key-frames), the algorithm is implemented recursively and its pseudo code is presented in Appendix B Section 1. This gives a set of walks each with a set of associated loops. Note that although loops may be nested such that one loop includes another, in our optimisation all possible nested sub-loops are included as illustrated in Figure 7.5.

7.2.2.2 Integer Linear Programming

Optimisation of Equation 7.26 is non-linear, however, it can be converted into constrained Integer Linear Programming (ILP) sub-problems. For a particular walk with loops \( L \), the corresponding number of repetitions \( n \) is optimised as four independent
Chapter 7. Surface Motion Graphs

ILP sub-problems, \( \rho = 1, 2, 3, 4 \).

\[
\begin{align*}
\text{minimise} & \quad C_\rho \cdot n_\rho \\
\text{subject to} & \quad n_{\rho,0} = 1 \\
& \quad 0 \leq n_i \leq +\infty, \text{ integer} \\
& \quad C_{d,\rho} \cdot n_\rho \geq 0 \\
& \quad C_{t,\rho} \cdot n_\rho \geq 0
\end{align*}
\]

(7.29a) (7.29b) (7.29c) (7.29d) (7.29e)

\( C_{d,\rho}, C_{t,\rho}, C_{s,\rho} \) and \( C_{p} \) are vectors representing the distance, time and total transition (smoothness) costs for each sub-problem \( \rho \), whose elements are computed as,

\[
\begin{align*}
C_{d,\rho,i} &= \text{sign}_d(\rho) \cdot w_d \cdot (f_d(l_i) - dv) \\
C_{t,\rho,i} &= \text{sign}_t(\rho) \cdot w_t \cdot (f_t(l_i) - tv) \\
C_{s,\rho,i} &= f_s(l_i) \\
C_{p,i} &= C_{d,\rho,i} + C_{t,\rho,i} + C_{s,\rho,i}
\end{align*}
\]

(7.30a) (7.30b) (7.30c) (7.30d)

where \( \text{sign}_d = (1, 1, -1, -1) \) and \( \text{sign}_t = (1, -1, 1, -1) \). For each sub-problem, \( n_{\rho}^{\text{opt}} \) is solved efficiently by a standard ILP solver. The optimal repeat times of loops \( n \) for a particular walk with loops \( l \) is then computed as the one that achieves the minimum combined cost,

\[
n^{\text{opt}} = \arg \min_{\rho=1,2,3,4} \{ C_{\rho} \cdot n_{\rho}^{\text{opt}} \}
\]

(7.31)

7.2.2.3 Graph Path Composition

Once the optimal walk and loops \( L^{\text{opt}} \) with repetitions \( n^{\text{opt}} \) have been evaluated, a complete path can be composed by \( F = n^{\text{opt}} \cdot L^{\text{opt}} \). The final motion sub-sequences are concatenated head-to-tail by matching the centroid of mesh at transitions. However, some loops may indirectly connect to the walk via other loops, e.g. \( l_2 \) via \( l_1 \) connecting to \( l_0 \) in Figure 7.5(b), if the repetition of connecting loops is zero, we cannot make the path. The feedback strategy is presented in Appendix B Section 2, when the isolation of loops happens, a constraint is added to the ILP solver. Finally, the graph path is concatenated as a character animation.
7.3 Experimental Results and Evaluation

3D character animations are synthesised from a database of 3D video [102] which comprises an actress (Roxanne) wearing three different costumes: a game character with shorts and t-shirt (Character1); a long flowing dress (Fashion1); and a shorter tight fitting dress (Fashion2). For each costume periodic (walk, run, stagger), motion transitions (walk to run, run to walk, walk to stand, stand to walk) and other motions (hit, twirl) are included. The captured 3D video sequences are unstructured meshes with unknown temporal correspondence and different mesh connectivity at each time-frame. A mesh contains approximately 100k vertices and 200k triangles.

Surface motion graphs are automatically constructed from 3D video sequences for the performer within three different costumes. Optimisation is performed in seconds for user-defined constraints on distance and time. Motions are concatenated without post-processing or blending to show the raw 3D video transitions. Synthesis results are presented in accompanying videos. An example of selected frames from a synthesised motion for Fashion1 captured in a virtual camera view is shown in Figure 7.6. These results demonstrate that the motion synthesis preserves the detailed clothing and hair dynamics in the captured 3D video sequences and does not produce unnatural movements at transitions.

Motion synthesis is evaluated for the three surface motion graphs which represent potential transitions with four pairs of key-frames for each costume as shown in Figure 7.8-7.10 and Table 7.1. Evaluation is performed by synthesising motions for target constraints on distance of \(1 - 20\)m in 1m intervals and times of \(1 - 40\)s in 1s intervals giving 800 sequences for each key-frame pair and 9600 synthesised sequences in total.

We consider a synthesis with traversing distance error less than 1 metre and time error less than 1 second as successfully meeting otherwise missing the target. This is because not all desired motion can be synthesised: first, not all desired motion is physical possible, e.g. it is not possible for a human walk or run 20 metres in 1 second; second, not all desired motion can be synthesised given a database in which only limited motion are captured. This has also explained missed target distributes in the region where target time is short while target distance is long shown in Figure 7.7-7.10. We also notice that for more complex motion which usually requires more time to perform, like Hit
or Twirl, the missed target region is larger than simpler motion, like Stand, Pose and Walk as shown in comparison between Figure 7.8-7.10 and Figure 7.7. Additionally, the database of Character 1 contains motion “Walk” and motion “Jog” while Fashion 1/2 only contain motion “Walk”. Figure 7.7-7.8 shows less missed targets than that in Figure 7.9-7.10. This can be explained by a larger 3D video database which contains more motions will increase the capability and flexibility of Surface Motion Graphs used for synthesising character animation. Therefore, for each key-frame pair, the maximum, minimum and root mean square errors over all synthesised sequences except those missing the target for distance moved and timing are presented in Table 7.1. This analysis shows that the maximum distance and timing errors are less-than 1% of the target indicating that the path optimization generates sequences which accurately satisfy the user-defined constraints. Smoothness cost is evaluated as a weighted average of Hausdorff Distance for overlapped individual frames at transitions. The weights are set to decrease about the central frame in the same way as $\chi(k)$ in Equation 7.10. Computation times are given for an ILP solver together with a Matlab implementation of the synthesis framework running on a single processor machine. The computation time is approximately constant with respect to the distance and timing constraints as indicated by the low standard deviation.

### 7.4 Discussion

There are some issues with Surface Motion Graphs needed be further clarified and discussed. First, the proposed Surface Motion Graphs is an analogue to traditional Motion Graphs. The main difference is that a traditional Motion Graph is built from a database of skeletal Motion Capture data, however, a Surface Motion Graph is from 3D video sequences which capture both the skeletal motion and surface dynamics. Since recovering the surface correspondence for 3D video sequences is still an open research problem, the way to identify optimal transitions for Motion Capture data (where the temporal correspondence is given) used by traditional Motion Graphs cannot be directly adopted for Surface Motion Graphs. The main contribution of the work is to avoid solving the surface correspondence problem but still be able to identify optimal transitions and build Surface Motion Graphs. Second, since the surface correspondence
7.4. Discussion

(a) Character 1: Stand#1 → Hit#45, $d_v = 10m$, $t_v = 10s$

(b) Fashion 1: Pose#1 → Twirl#85, $d_v = 10m$, $t_v = 20s$

(c) Fashion 2: Pose#1 → Twirl#100, $d_v = 10m$, $t_v = 20s$

Figure 7.6: Example meshes for synthesised motion from 3D video database of Roxanne. We denote frame no. as #. (a) Character1: Stand#1 → Hit#45 with traverse distance 10 metres and traverse time 250 frames; (b) Fashion1: Pose#1 → Twirl#85 with traverse distance 10 metres and traverse time 500 frames; (c) Fashion2: Pose#1 → Twirl#100 with traverse distance 10 metres and traverse time 500 frames.

is remained unsolved, frames in transitions are not blended and two motion are concatenated by switching at their central frame of transition. This may still introduce some discontinuities at transitions which partially explains in some cases there are some visual artefacts at transition although in practice most synthesized motions look reasonable good. In future work, the blending at transition needs to be solved either by solving the surface correspondence or finding a blending scheme without requirement of surface correspondence, like time-space warping. Third, although the implications of decomposing all possible paths and cycles in a Surface Motion Graph will give the global optimal solution but it is computational complex which limits the scalability over
larger database. In future, more efficient graph search algorithms will be investigated and incorporated into current Surface Motion Graphs framework. Finally, the evaluation of Surface Motion Graphs is currently more like a qualitative evaluation than a quantitative one. Although the minimum and maximum of smoothness are measured in centimeters and distance and time error are measured in meters and seconds, due to the lack of ground-truth, how similar the synthesized motion is to a real one cannot be told. In future, the way to evaluate Surface Motion Graphs quantitatively will be investigated.

7.5 Summary

In this chapter, we have presented a framework for concatenative human motion synthesis from 3D video sequences according to user-defined constraints on movements, distance and timing. Transitions between 3D video sequences are identified without the requirement for temporal correspondence using 3D shape similarity over an adaptive temporal window of frames. A surface motion graph is automatically constructed to represent potential transitions for both cross-transitions between different motion sequences and self-transitions for cyclic motion. Path optimisation is performed between user-specified key-frames using a standard ILP solver to satisfy constraints on distance and timing with repeated motions for loops in the graph. Results demonstrate that concatenative synthesis of novel sequences accurately satisfies the user constraints and produces motions which preserve the detailed non-rigid dynamics of clothing and loose hair. This approach greatly increases the flexibility in the reuse of 3D video sequences allowing specification of high-level user constraints to produce novel complex 3D video sequences of human motion.
<table>
<thead>
<tr>
<th>SMG: Key-frames</th>
<th>Smooth. (cm)</th>
<th>Distance error (m)</th>
<th>Time error (sec.)</th>
<th>Cputime (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>max</td>
<td>min</td>
<td>rms</td>
</tr>
<tr>
<td><strong>Character 1:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stand#1 → Hit#45</td>
<td>4.2</td>
<td>19</td>
<td>0.0001</td>
<td>0.17</td>
</tr>
<tr>
<td>Stand#1 → Walk#16</td>
<td>4.2</td>
<td>26</td>
<td>0.0002</td>
<td>0.21</td>
</tr>
<tr>
<td>Walk#16 → Jog#13</td>
<td>4.2</td>
<td>26</td>
<td>0.0001</td>
<td>0.21</td>
</tr>
<tr>
<td>Jog#13 → Hit#45</td>
<td>4.2</td>
<td>25</td>
<td>0.0002</td>
<td>0.20</td>
</tr>
<tr>
<td><strong>Fashion 1:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose#1 → Twirl#85</td>
<td>5.8</td>
<td>20</td>
<td>0.0001</td>
<td>0.23</td>
</tr>
<tr>
<td>Pose#1 → Walk#15</td>
<td>5.8</td>
<td>20</td>
<td>0.0001</td>
<td>0.47</td>
</tr>
<tr>
<td>Walk#15 → WalkPose#37</td>
<td>4.2</td>
<td>13</td>
<td>0.0000</td>
<td>0.33</td>
</tr>
<tr>
<td>WalkPose#37 → Twirl#85</td>
<td>4.2</td>
<td>20</td>
<td>0.0001</td>
<td>0.29</td>
</tr>
<tr>
<td><strong>Fashion 2:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose#1 → Twirl#100</td>
<td>4.4</td>
<td>21</td>
<td>0.0000</td>
<td>0.23</td>
</tr>
<tr>
<td>Pose#1 → Walk#15</td>
<td>4.8</td>
<td>18</td>
<td>0.0008</td>
<td>0.39</td>
</tr>
<tr>
<td>Walk#15 → WalkPose#37</td>
<td>4.4</td>
<td>16</td>
<td>0.0002</td>
<td>0.36</td>
</tr>
<tr>
<td>WalkPose#37 → Twirl#100</td>
<td>4.4</td>
<td>18</td>
<td>0.0002</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Table 7.1: Evaluation for Roxanne. A grid of target 20 x 40 (metres x seconds) is tested for each pair of key-frames shown in the first column.
Figure 7.7: Evaluation for synthesised motion from 3D video database of Roxanne (Character1): Key-frames are set as 1st frame of the motion “Stand” and 30th frame of “Walk2Stand” (Character 1: Stand#1 → Walk2Stand#30). Target traversing distance and time are shown as a grid of cross. A green cross denotes a met target and a cyan cross a missed target. The error between a met target and corresponding synthesis is denoted as a red line, the longer line the bigger the error.
Figure 7.8: Evaluation for synthesised motion from 3D video database of Roxanne (Character1): Start-end key-frames are set as 1st frame of the motion “Stand” and 45th frame of “Hit” (Character 1: Stand#1 → Hit#45). A green cross denotes a met target and a cyan cross a missed target. The error between a met target and corresponding synthesis is denoted as a red line, the longer line the bigger the error.
Figure 7.9: Evaluation for synthesised motions (Fashion1): Key-frames are set as 1st frame of motion “Pose” and 85th frame of “Twirl” (Fashion 1: Pose#1 → Twirl#85). Target distance and time are shown as a grid of cross. A green cross denotes a met target and a cyan cross a missed target. The error between a met target and corresponding synthesis is denoted as a red line, the longer line the bigger the error.
7.5. Summary

Figure 7.10: Evaluation for synthesised motion from 3D video database of Roxanne (Fashion2): Key-frames are set as 1st frame of the motion “Pose” and 100th frame of “Twirl” (Fashion 2: Pose#1 → Twirl#100). Target traversing distance and time are shown as a grid of cross. A green cross denotes a met target and a cyan cross a missed target. The error between a met target and corresponding synthesis is denoted as a red line, the longer line the bigger the error.
Chapter 8

Conclusions

State-of-the-art production techniques can create visually stunning synthetic imagery. However, animated content is not necessarily believable. Computer generated images lack the detailed changes in shape and appearance that we experience everyday and see in conventional film and video. Among all kinds of animations character animation or human motion is of the greatest interest. Animation of human motion is difficult due to our ability to perceive natural motion. Motion capture appears to help create such natural looking motions and has become a standard production technique. Extensive research has addressed the problem of processing and re-using motion capture for both off-line and on-line character animation production. However, traditional motion capture techniques for the articulation of the human skeleton cannot capture surface dynamics such as hair or loose clothing motion, therefore, truly realistic synthetic human motion still requires a lot of human intervention and manual work which is time-consuming and high cost.

In the last decade, full 3D video-based scene capture of people, called surface motion capture, has advanced to the stage of capturing both shape and appearance which is missing from marker-based motion capture. This holds the potential to create truly realistic animated content. Currently the captured 3D video is difficult to re-use other than for replay which largely limits the application. On the other hand, example-based synthesis has now been introduced as a means to record and replay the detailed dynamics of a scene. Previous work has addressed the re-use of 2D video, limited viewpoint 2.5D video and manually constructed animation from 3D video. No work to-
date has addressed automated animation and re-use of a complete dynamic 3D model of a person. This thesis has provided a framework to address the general problem of constructing automatic and user-directed animation sequences of 3D surface capture data or 3D video to produce highly realistic computer graphics models of people. In proposed work, we have found:

- **Shape Histograms** outperforms Shape Distribution, Spin Image and Spherical Harmonics Representation when identifying similar poses of the same person performing different motions in 3D video sequences.

- An adaptive temporal filtering used to identify the optimal transitions between motions is with a decent quality and relatively a low additional computational cost.

- The switch at central frame of transitions when concatenating two motions, in practice, can still produce reasonable smooth transitions resulting visually satisfied motion synthesis.

- **Realistic looking synthesized motion** including walking, running, twirling and hitting etc. can be produced in current framework of Surface Motion Graphs.

### 8.1 Contributions

The main difference between conventional motion capture data and 3D video, specifically, surface motion capture (SurfCap) [102] is that the former has a hierarchical model with known temporal correspondence while the latter is model-less without temporal surface correspondence. Although many researchers have tried to solve temporal surface correspondence for surface motion capture, this remains an open problem due to the quality and reliability. This prevents direct application of methods to reuse conventional motion capture data for animation production to surface motion capture data or 3D videos. This thesis introduces a framework, Surface Motion Graphs, which does not require temporal surface correspondence for 3D videos but is able to reuse the data to produce novel animations analogous to motion graphs previously introduced to reuse skeletal Motion Capture data [64].
In order to allow production of novel animations from 3D video sequences the following contributions have been made in this work:

- A quantitative evaluation of shape similarity metrics for 3D video sequences of people has been presented. 3D shape descriptors including Shape Distribution [91], Spin Image [58], Shape Histogram [3] and Spherical Harmonics[62] are compared against a synthetic ground-truth dataset of temporal surface sequences of people in which surface correspondence is predefined. Evaluation shows that a volume-sampling Shape Histogram (SHvr) obtains the highest ROC performance in the task of matching frames of time-varying 3D mesh sequences of people with similar global shape without a prior model or temporal correspondence [52].

- Introduction and a quantitative evaluation of temporal shape similarity metrics for 3D mesh sequences of people have been presented. Two novel 4D shape-flow descriptors, global Shape-flow Histograms (SHvrG) and local Shape-flow Histograms (SHvrS), are introduced to match frames of time-varying 3D mesh sequences of people with similar global shape and motion. Shape-flow and temporal filtered 3D shape descriptors are compared against a synthetic ground-truth dataset of temporal surface sequences of people in which surface correspondence is predefined. Evaluation shows that a time-filtered volume-sampling Shape Histogram (SHvrT) provides the best trade-off between computational complexity and matching accuracy [53, 51].

- A novel shape-colour similarity metric for 3D video has been introduced and quantitatively evaluated. The novel shape-colour descriptor, Shape-Colour Histograms, is introduced to match frames of 3D video sequences of people with similar shape and appearance. Time-filtered shape-colour and shape-only descriptors are compared against a synthetic ground-truth dataset of 3D video sequences of people in which surface correspondence is predefined. Evaluation shows that the time-filtered Shape-Colour Histograms accurately matches similar 3D video frames in terms of shape, appearance and motion [48].

- A method for automatic extraction of key-frames from 3D video has been introduced. The approach analyses the structure of the self-similarity matrix to
provide a method for 3D video summarisation. The method trades off rate (compactness) against distortion (faithfulness) to obtain the optimal set of key-frames. The approach is demonstrated to give improved performance for summarisation of 3D video sequences of people over previous techniques [49].

- An automatic method for concatenative animation from 3D video sequences has been presented, Surface Motion Graphs (SMG). This approach is analogous to skeletal Motion Graphs previously used for conventional motion capture but takes into account the full surface shape similarity. The method incorporates similarity metrics and key-frames extraction together enabling user-controlled production of 3D character animation. An Integer Linear Programming approach is introduced to find the optimal path between start and end-key frames according to user-defined constraints and maximize the transition smoothness. SMGs are demonstrated to produce novel animations from captured 3D video sequences which preserve the realistic dynamics of shape and appearance. The process of graph construction is off-line but path optimisation is online. Consequently a user can produce many different animations according to different high-level constraints from a single SMG [50, 47].

8.2 Limitations

There are several limitations of the proposed work:

- **Smooth transitions.** Motions are currently concatenated by switching at the central frame of the transition. In this way, we cannot guarantee that the transition is truly smooth although in practice it works well in most cases and does not introduce visual artefacts.

- **Motion on a curve.** Motions in a straight line are currently synthesized and the bending along an arbitrary curve on the ground is problematic. This is because when bending happens the character's two feet are both on the ground, there is no way to concatenate without blending and produce a nice and smooth animation without introducing foot sliding or "jump" visual artefacts.
• **Timing and distance.** The presented system cannot change the timing of captured sequences which limits its application: to produce a desired animation the same speed or timing of captured motion is required. The presented system cannot produce animation that exactly satisfies user-specified tranverse distance and time.

• **Motion manipulation.** The presented system does not have the function to do motion manipulation, for example, a hit motion with the freedom to change the hitting point.

• **Scalability.** The main scalability limitation exists in the optimal graph path search scheme. Identifying all paths and loops over a given Surface Motion Graph will be prohibitively computational complex as the SMG becomes larger. Therefore, a more efficient graph search algorithm must be provided when the 3D video database goes larger. The algorithm does not have to find the global optimal path like currently our algorithm does, a reasonable sub-optimal path is acceptable.

• **Interactive control.** There is currently no interactive control when producing the desired character animation in presented system.

### 8.3 Future Work

To solve all of these limitations, future work is summarised as follows:

• **Time-Space warping.** To produce smooth transitions, to synthesize motion following an arbitrary curve, to make the exact the timing\&distance and to manipulate motion, a method of space-time warping is desired. There are two options to do time-space warping: with and without temporal surface correspondences. Although with temporal surface correspondences, the warping will be easier, to solve correspondences itself is difficult and it is still an open research problem. On the other hand, one of the fascinating features of the presented system is that it does not require solving correspondences. Therefore, in the future work it is desirable either to introduce methods of space-time warping which do not require accurate surface correspondence or solve surface correspondence for transitions.
• **Scalability** A more efficient graph search algorithm will be investigated and the goal is to find reasonable sub-optimal path in a reasonable time instead of finding the global optimal one. And the 3D video database will grow larger to provide more possibility and flexibility when synthesizing the motion.

• **Interactive control.** Although the pre-processing of 3D video data is off-line including extracting 3D shape-only or shape-colour descriptors for each frame and the construction of surface motion graph, the path optimization to satisfy user-defined constraints is an online process. Therefore, the system has the potential to provide the user the interactive control over the production of animation. Interactive control is desired because it allows creative professionals to iteratively produce animation and change it accordingly, in a familiar, to produce a high-quality animation for film applications. This is also interesting because it could allow amateur users to interactively control the character during a game with the added realism of 3D video.
Appendix A

ROC performance for Static and Temporal Shape Descriptors
Figure A.1: Evaluation of static shape descriptors against Temporal Ground Truth (TGT). ROC performance for 28 motions across 14 people.
Figure A.2: Evaluation of temporal filtered shape descriptors with a fixed window size $5 (N_t = 2)$ against Temporal Ground Truth (TGT). ROC performance for 28 motions across 14 people.
Figure A.3: Evaluation of shape-flow descriptors with a fixed window size 5 ($N_t = 2$) against Temporal Ground Truth (TGT). ROC performance for 28 motions across 14 people.
Appendix B

Algorithms for Surface Motion

Graphs
B.1 Find All Walks and Loops by DFS

```plaintext
find_all_walks(adj, source, sink, walk) walk ← source;
if source = sink then
    walks = walk;
    return;
else
    walks = [];
    foreach node in source's neighbours do
        if node not in walk then
            new_walks = find_all_walks(adj, node, sink, walk);
            walks ← new_walks;
        else
            loop = [];
            i = index of node in walk;
            loop ← walk(i:end);
            if loop not in loops then
                loops ← loop;
            end
        end
    end
end
```

Algorithm 1: Find all walks and loops by DFS

B.2 Feedback Strategy

```plaintext
1 := 1..N;
n ← Solve ILP;
k = 1;
while exist i ∈ I, n_i ≥ 1 and l_i not connected to l_0 do
    set a 2D matrix A according to n:
    foreach i ∈ I do
        if n_i = 1 then
            a_k,i = 1;
        else
            a_k,i = 0;
        end
    end
    add constraints A_n ≥ 1 to ILP solver;
n ← Solve ILP;
k = k + 1;
end
```

Algorithm 2: Feedback strategy


[58] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in


[69] I. S. Lim and D. Thalmann. Key-posture extraction out of human motion data by curve simplification.


[79] T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara. Real-time 3d shape reconstruction, dynamic 3d mesh deformation,


