Computational Models of Socially Interactive Animation

Dumebi Okwechime

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, U.K.

February 2011

This thesis is dedicated to the loving memory of my dad
Augustine Dibia Okwechime

Abstract

The aim of this thesis is to investigate computational models of non-verbal social interaction for the purpose of generating synthetic social behaviour in animations. To this end, several contributions are made: A dynamic model, providing multimodal control of animation is developed and demonstrated using various data formats including motion capture and video; A social interaction model is developed, capable of predicting social context/intent such as level of interest in a conversation; and finally, the social model is used to drive the dynamic model, which animates appropriate social behaviour of a listener in a conversation in response to a speaker.

A method of reusing motion captured data by learning a generative model of motion is presented. The model allows real-time synthesis and blending of motion, whilst providing it with the style and realism present in the original data set. This is achieved by projecting the data into a lower dimensional space and learning a multivariate probability distribution of the motion sequences. Functioning as a generative model, the probability density estimation is used to produce novel poses, and pre-computed motion derivatives combined with gradient based optimisation generates the animation.

A new algorithm for real-time interactive motion control is introduced and demonstrated on motion captured data, pre-recorded videos and HCI. This example-based method uses the original motion data for synthesis by seamlessly combining various subsequences together. A novel approach to determining transition points is presented based on k-medoids, whereby appropriate points of intersection in the motion trajectory are derived as cluster centres. These points are used to segment the data into smaller subsequences. A transition matrix combined with a kernel density estimation is used to determine suitable transitions between the subsequences to develop novel motion. To facilitate real-time interactive control, conditional probabilities are used to derive motion given user commands. The user control can come from any modality including auditory, touch and gesture. The system is also extended to HCI using audio signals from speech in a conversation to trigger non-verbal responses from a synthetic listener in real-time. The flexibility of the method is demonstrated by presenting results ranging from data sets composed of vectorised images, 2D and 3D point representations.

In order to learn the dynamics of social interaction, experiments are conducted to elicit natural social dynamics of people in a conversation. Semi-supervised computer vision techniques are then employed to extract social signals such as laughing and nodding. Learning is performed using association rule data mining to deduce frequently occurring patterns of social trends between a speaker and listener in both interested and not interested social scenarios. The confidence values from rules are utilised to build a Social Dynamics Model (SDM), that can then be used for both classification and visualisation. By visualising the rules generated in the SDM, analysing distinct social trends between an *interested* and *not interested* listener in a conversation is possible. The *confidence* values extracted from the mining can also be used as conditional probabilities to animate social responsive avatars. A texture motion graph is combined with the example-based animation system developed earlier within the thesis. Using the mined rules of social interaction, social signals are synthesised within the animation, providing the user with control over who speaks and the interest level of the participants.

Key words: Human Computer Interaction, Probability Density Function, Motion Synthesis, Character Animation, Modelling Social Interaction, Social Behavioural Analysis, Apriori Mining.

Email: d.okwechime@surrey.ac.uk

WWW: http://info.ee.surrey.ac.uk/Personal/D.Okwechime/

Acknowledgements

Firstly, I would like to extend my gratitude to EPSRC (Engineering and Physical Sciences Research Council) for the Doctoral Training Grant that funded this research.

I would like to thank Professor Richard Bowden, for his encouragement, dedication, and support throughout my undergraduate and postgraduate studies. Undertaking an undergraduate final year project under his supervision motivated my interest in Ph.D research, which he subsequently supervised, making it an enjoyable, successful, and memorable experience. I am eternally grateful.

To my fellow colleagues in CVSSP, especially Bud Goswami, Andrew Gilbert, and Helen Cooper. Thank you for all the assistance you gave me during my Ph.D, and for providing a pleasant and supportive environment to work in. A special thanks to Eng-Jon Ong for lending me his invaluable insight and experience. My discussions with him helped stimulate and inspire ideas which kept me enthusiastic and motivated.

To my examiners, Dr Nadia Bianchi-Berthouze and Dr John Collomosse. Thank you for your constructive and insightful evaluations and suggestions for my thesis.

To staff at the university, especially James Field and those who work in the Faculty of Engineering and Physical Sciences. Thank you for all your efforts and hard work, and for keeping the university and the computer vision department running smoothly.

Finally, I would like to extend a tremendous thanks to my family and closest friends. Especially my dad and mum, Augustine and Rose Okwechime, my sisters, Lolita Ejiofor and Nkechi Okwechime, and my second half, Jania Aghajanian. They stood by me during my studies, and without their love, constant support and encouragement, this Ph.D would not have been possible.

Contents

List of Figures

List of Tables

Nomenclature

MIMiC	Multimodal Interactive Motion Controller
SDM	Social Dynamics Modal
GMM	Gaussian Mixture Modal
PDF	Probability Density Function
PCA	Principal Component Analysis
fps	frames per second
Real Time	At more than 25 fps
MoCap	Motion Capture
MFCC	Mel-Frequency Cepstral Coefficient
RGB	Red Green Blue Colour Space
HCI	Human Computer Interaction
STD	Standard Deviation
VAR	Variance
FFT	Fast Fourier Transform
HTK	Hidden Markov Model Toolkit
LSQ	Least Square
SD	Standard Definition
2D	Two-Dimensional
3D	Three-Dimensional
DOF	Degree of Freedom

Symbols

R	Number of markers
D, d	Dimensionality
(x, y, z)	3D cartesian coordinates
$\mathbf{S}$	Sample convariance matrix
t	Time in unit of frames
T	Transpose
σ	Standard deviation
max	Highest likelihood pose
$max\Delta$	Highest likelihood movement
Γ	Noise Term
γ	Normally distributed random number
ψ	Noise control parameter
$\mathbf{X}$	Motion Sequence
$\mathbf{x}_i$	i^{th} vectorised frame
N_T	Total number of frames
$\mathbf{Y}$	Set of all points dimensionally reduced
$\mathbf{y}_i$	i^{th} dimensionally reduced data point
$\mathbf{V}$	Eigenspace projection
$\mathbf{T}$	Eigenvectors
$\boldsymbol{\lambda}$	Eigenvalues
$\boldsymbol{\mu}$	Sample mean
P	Probability
N_n	Number of nearest neighbours
$\mathbf{Y}'$	Set of points containing nearest neighbouring kernels
$p(\mathbf{y}_i)$	A kernel
$G(\mathbf{y}_i, \Sigma)$	Gaussian centred on a data example
$\sqrt{\boldsymbol{\lambda}}$	Standard deviation of mode
α	Width control variable
N_c	Number of k-medoid points/cut point clusters
$\mathbf{Y}_n^c$	n^{th} set of cut points
$\mathbf{y}_n^c$	Cut point member of the n^{th} cluster
$\boldsymbol{\delta}_n^c$	K-medoid point
θ	Threshold
Q_n	Number of cut points in the n^{th} cluster
$\mathbf{z}_n^c$	End transition point
C_n^z	Index of the cluster $\mathbf{z}_n^c$ belongs to
$p_{k,l}$	Probability of going from cluster k to cluster l
$\mathbf{P}$	Transition Matrix
C_t	Index for cluster/state at time t

Q_t	Number of possible transitions from $\mathbf{y}_t^c$
m	Index for possible transitions
$\mathbf{\Phi}$	Set of likelihood for each transition
$\boldsymbol{\phi}_i$	i^{th} likelihood
k	Index of newly chosen end transition point
r	Random number between 0 and 1
u	Number of trellis levels
N_s	Number of input symbols for quantisation
N_r	Number of training examples associated to a symbol
E_r	Set of training examples
e_i	A training example
$p_{q,r}$	Mapping of the r^{th} input symbol to the q^{th} cut point cluster
$input_r$	r^{th} input symbol
W	Weight
N_I	Total number of instances
F	Set of *instances*
$\boldsymbol{\mathsf{f}}$	Binary vector of active and inactive signals
R_i^A	Set of social signals of the speaker
R_i^C	Set of social signals of the listener
r_i^A	Speaker's social signal
r_i^C	Listener's social signal
sup	Support value
$conf$	Confidence value
R	Set of all extracted rules
s	Temporal bagging window size
ι	Frame index
η	Ratio threshold
SDM_{int}	Interested classifier
SDM_{not}	Not interested classifier
EL	Eliminated Listener
TS	Trained Speaker
MAX	Maximum window size
$Incr$	Query window size
$StartPt$	Window starting frame
$FrWin$	Query Window
$SetR$	Set of association rule within window
$Score_h$	Binary vector for pass or fail of prediction
$Prediction_v$	Overall percentage of predictions
[V]	Voiced
[T]	Talking
[L]	Laughing
[S]	Head shakes
[N]	Nod
[A]	Activity measure
[G]	Gaze direction
[GL]	Gaze left
[GR]	Gaze right
[GC]	Gaze centre
[G-S]	Gaze speaker
[G-OL]	Gaze other listener
[G-N]	Gaze no one

List of Publications

Elements from this manuscript have appeared in several publications in the field of computer graphics, multimedia, HCI, computer vision, and behavioural science. The resulting publications are as follows:

1. Okwechime D, Bowden R, 'A Generative Model for Motion Synthesis and Blending Using Probability Density Estimation.' *In Fifth Conference on Articulated Motion and Deformable Objects (AMDO'08)*, Mallorca, Spain, Jul. 9-11 2008, pp. 218 - 227. **(Chapter 3)**

2. Okwechime D, Ong E. J, Bowden R, 'Real-Time Motion Control Using Pose Space Probability Density Estimation.' *In Proceedings of the 12th International Conference on Computer Vision (ICCV'09): IEEE International Workshop on Human-Computer Interaction (HCI'09)*, Kyoto, Japan, Sep. 29 - Oct. 2 2009, pp. 2056 - 2063. **(Chapter 4)**

3. Okwechime D, Ong E. J, Bowden R, 'MIMiC: Multimodal Interactive Motion Controller.' *Accepted for publication In IEEE Transactions on Multimedia (TMM)*, 2011. **(Chapter 3,4)**

4. Okwechime D, Ong E. J, Gilbert A, Bowden R, 'Visualisation and Prediction of Conversation Interest through Mined Social Signals.' *In Proceedings of the 9th IEEE Conference on Automatic Face and Gesture Recognition (FG'11): IEEE International Workshop on Social Behavior Analysis*, Santa Barbara, CA, Mar. 21-25 2011. **(Chapter 5)**

5. Okwechime D, Ong E. J, Gilbert A, Bowden R, 'Social Interactive Human Video Synthesis.' *In Proceedings of the 10th Asian Conference on Computer Vision (ACCV'10)*, Queenstown, New Zealand, Nov. 8-12 2010. **(Chapter 6)**

6. Okwechime D, Ong E. J, Gilbert A, Bowden R, 'Modelling Socially Interactive Avatars.' *In review process of IEEE Transactions on Systems, Man, and Cybernetics, Part B (TSMC)*, 2011. **(Chapter 5,6)**

Chapter 1

Introduction

Motion synthesis has been an emerging topic in recent years, highly applicable to the movie and gaming industries. A popular application of motion synthesis is in the reuse of motion capture data for the purpose of generating computer animations. Earlier techniques relied on interpolation to generate intermediate frames from manually defined key-frames. This method offers high controllability to the animator but is very time consuming and requires a highly skilled animator to produce realistic animations. Realism is of great importance to the entertainment industry because it brings to life computer animated characters, making their performances enjoyable and believable. The human visual system can easily recognise characteristic motion but it is particularly sensitive to errors, repetitions, or discontinuities in the motion. As a consequence, in order to generate animations that look realistic, it is necessary to develop methods to capture, maintain and synthesise intrinsic style to give authentic realism to motion data.

Motion capture provides a cost effective solution to realism, as life-like motion is easily acquired and large libraries of motion are available for use. Figure 1.1 shows an example of a motion captured walk sequence. To provide realistic results, multiple sequences need to be blended together resulting in a seamless and life-like animation. Furthermore, recorded animation lacks the variability of natural motion and repeating a motion can look false due to its lack of natural variance. Within this thesis, a *generative model* is explored to synthesise and blend between different cyclic human

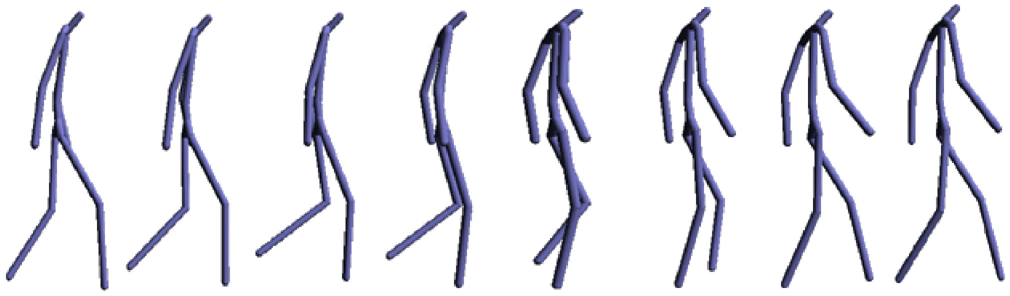


Figure 1.1: **Example MoCap:** Example motion captured data of a walk sequence

articulated motion, resulting in sequences of realistic actions which would otherwise be difficult to achieve using keyframing. Natural variation is also incorporated into the model, allowing repetitive motion to appear slightly different each time, as one would expect in real life.

As well as motion capture synthesis, temporal texture synthesis from photorealistic motion in video, is a vital tool in film production. Figure 1.2 shows examples of temporal textures in video of stochastic and non-stachastic motion. (A) and (B) are stochastic videos of a candle flame and plasma beam respectively undergoing motion, and (C) is a non-stochastic video of person's facial expression when engaged in conversation.

When filming a movie, certain elements in a video scene such as the movement of trees blowing in the wind, do not perform on cue. It may not always be cost effective, safe or even possible to control the surroundings to match the director's intentions. Likewise, the ability to control the movement of a character in a video scene can provide an attractive alternative to post-production filming, providing movie editors with the means to edit an actor's performance without having to re-record the scene, which could also be very expensive and time consuming. Although CGI (Computer Generated Imagery) is the vastly popular medium for computer game characters and post-production modifications, photo-realistic temporal textures has proven to heighten realism especially in the gaming experience [44]. This thesis presents a novel approach to generating animations from video. An *example-based* motion model, whereby the original motion data is retained to use in synthesis (as opposed to a generalisation) is adopted, resulting in no loss in motion detail and increased realism. This model

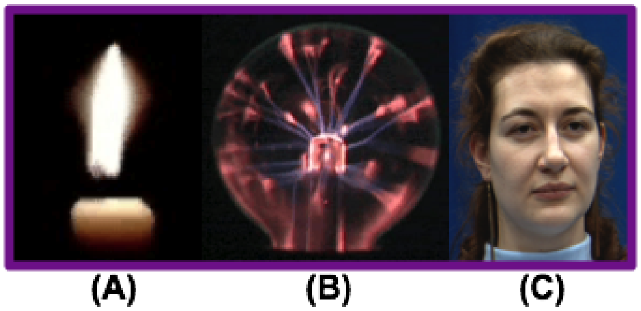


Figure 1.2: **Example temporal textures:** Example temporal textures in video of stochastic and non-stachastic motion. (A) and (B) are stochastic videos of a candle flame and plasma beam respectively, and (C) is a non-stochastic video of a person's facial expression when engaged in conversation.

is unique in that it is capable of handling large data sets due to its novel motion segmentation approach. It is also not tailored to animate from video textures only, but applicable to virtually any motion format. As well as textures, 2D, and 3D (MoCap) point representations are demonstrated.

In HCI (Human Computer Interaction), real-time interactive control is of vital importance when developing effective user interfaces. For instance, when operating a computer, one utilises a mouse and a keyboard for interaction. Using these interfaces, commands and intentions are translated by the machine and immediately executed. As society becomes more technologically adept and advanced, human and computer interfaces are increasingly becoming more intuitive. From the use of motion sensors to turn on lights and open doors (as opposed to light switches and door handles), to fully automated telephone systems with voiced command capabilities (not requiring keypads), to the next generation portable devices with multiple touch screens, gyroscope sensors to determine landscape or portrait visualisation, and so forth. As such, real-time interactive control of motion data could prove valuable in providing a human user with an effortless interface for interacting with animations.

Intuitive interfaces to control motion data are difficult because motion data is intrinsically high dimensional and most input devices do not map well into this space. Mouse and keyboard interfaces can only give position and action commands, so an autonomous approach is needed to translate user commands to appropriate behaviours and transitions in modelled motion data. In this thesis, by learning the mapping between motion subspaces and external stimulus, a multimodal motion controller is developed giving the user real-time interactive *Multimodal Control* of the creation of novel sequences. The external stimulus can come from any modality and is demonstrated within this work using auditory, touch and gesture.

Although there has been a vast amount of research in the field for motion capture animations and human video texture synthesis, little work has been done in developing socially interactive avatars, capable of responding appropriately to non-verbal communication. To make this possible, a method of modelling social dynamics in natural conversation is needed.

As naturally social entities, humans can easily extract social information from non-verbal communication without the need of understanding what is being said. Psychologists believe this skill is hard-wired in the human brain [67]. Gesture, vocal signal, and body language triggers unconscious analysis of socially relevant information [8]. Since non-verbal communication plays such an important role in our social interaction, a method of modelling it would prove valuable in understanding our relationships, identifying context/intent, or generating synthetic responses in an Artificial Intelligent (AI) context.

The ability to build models of social dynamics could assist social scientists and medical psychologists with diagnosing social related conditions from just a short period of video observation. It could enhance a machine's understanding of human social behaviour, resulting in better man and machine interfaces, and could also be used to drive animation resulting in socially interactive avatars.

Both in movies and 2D/3D animations, a crowded social environment such as a busy restaurant scene, would be a challenging setting to direct or animate. This is mainly due to the vast amount of characters involved. Using autonomous socially interactive avatars would make this complex and potentially expensive task easier. By selecting the avatars based on their social characteristics such as their level of interest in conversations, their politeness or aggressiveness, or even their emotional attachment to each other, a crowded social scene can be played out without the need for animating or directing each individual character.

Several gaming platforms are exploring new ways of enhancing the gaming experience, introducing innovative game controllers that allow users to better interact with their games. The idea of building a relationship with a video game character is not a new one, starting as early as 1996 with the Japanese handheld digital pets called Tamagotchi. Now, with the advancement in gaming technology, interacting and exchanging social signals with an interactive avatar would be a step towards true virtual relationships.

The focal point of this thesis addresses the research question:

> *Can a social model be used to drive a motion synthesis model, and generate realistic autonomous social behaviour in animations*

To this end, a model for non-verbal communication is devised, which also allows classification and visualisation of multimodal exchanges in social signals between a speaker and listener in a conversation. This social model is then applied as an autonomous social-context controller for human video motion synthesis. However, unlike social models that rely on intangible psychological observations, the approach adopted in this work uses tangible rules governed by the data to discern distinct trends and characteristics.

This thesis is divided into the following chapters. Chapter 2 reviews related work in the field of motion synthesis and social behaviour analysis, highlighting their main contributions and limitations.

This thesis consists of a number of key contributions presented over four technical chapters. A statistical model for motion synthesis and blending is first presented in Chapter 3. By modelling motion as a *probability density function* (PDF), the model can synthesise novel motion whilst retaining the natural variance inherent in the original data. Blending is as a result of linearly interpolating between different PDFs.

Chapter 4 extends this generative approach to an example-based motion model, whereby the original data itself is used for synthesis. By combining this motion model with an interactive *multimodal controller*, a *Multimodal Interactive Motion Controller* (MIMiC) is developed, giving a user multimodal control of motion data of various formats.

Chapter 5 introduces the *social dynamic model* (SDM), capable of accurately predicting conversational interest using mined social signals. Chapter 6 then describes how this social model is used to intuitively drive animation resulting in autonomous socially interactive avatars. This thesis then ends with conclusions in Chapter 7, where the achievements and limitations of this work is discussed with potential improvements and developments for future work.

Chapter 2

Literature Review

The work within this thesis is a combination of two disciplines; *Motion synthesis* for the purpose of generating computer animations, and *human social analysis* with the aim of understanding and modelling natural human social behaviour. As explained earlier, the focus of this research is to use the human social model to autonomously drive the motion model, resulting in socially interactive animations. This motion model can synthesise motion in various formats. Within this work, this is demonstrated on motion capture data, and temporal textures (RGB pixels from video). Additionally, as well as deriving social animations, the human social model can also efficiently predict social context from a short period of video observation.

To this end, this chapter starts with a literature review on techniques used for motion synthesis on *motion capture* and *temporal textures in video* respectively. This is then followed by a review on related research in *social behaviour analysis*. Finally, this chapter concludes with a summary of the literature reviewed, highlighting the open areas of research this thesis addresses.

2.1 Motion Capture Synthesis

Motion capture technology allows one to digitally capture and record the 3D movements of a performer in a computer. The recordings are of high quality and can reproduce an accurate computer model of the performer's movement which would otherwise be difficult to obtain with traditional methods of animation.

The motion capture system consists of a performer wearing markers on their body to identify motion by the position and orientation of the markers. The motion capture system records the positions, angles, velocities, accelerations and impulses to a computer, providing an accurate digital representation of the motion.

It started as a tool for choreographic study and for clinical assessment of movement in biomechanics research. It expanded into education, training, and more recently computer animation for cinema and video games [27]. In the early 1880s, Muybridge [86] produced one of the earliest studies of biomechanics, which started by simply observing series of photographs of animal locomotion. This led to the development of point-light displays attached to the joints of a person's body in order to gather information of a person's movement [124]. With technological advancements, the more frequently used motion capture systems are based on mechanical, magnetic and optical sensors. The mechanical system, which is the older of the group, consisted of a performer wearing mechanical armatures and encoders, heavily restricting the performer's movement. An improvement was offered by the magnetic system which utilises magnetic sensor, however the drawback to this technology is its sensitivity to metal in the capturing area which introduces noise into the final data [95]. The optical system consisting of reflective markers is an approach highly invested in Hollywood productions today. It provides high sampling rate and accuracy, but several problems can occur during capture such as marker occlusion and false reflections. This means that recordings must be post-processed which can be tedious and time-consuming.

Motion capture is costly, however, it is a faster and more reliable method of producing realistic animation compared to traditional key-framing techniques. Its main limitation is once the data has been collected it is difficult to edit without losing the natural qualities of the movement. Outlined in this section are various techniques developed to allow pre-recorded motion captured data to be edited, re-used, and blended to create novel motion. These are grouped according to the similarity of their techniques and in their applications.

Constraint-based methods [136, 49, 75, 76] generalise user specified kinematics and configuration constraints into an optimisation problem which is used to modify motion clips. Signal Processing techniques [137, 26, 126, 125, 118] represent human motion as a time varying signal, which is partitioned into various frequency components in order to apply global transformation on existing motion.

A common approach is using interpolation and blending techniques [102, 103, 112, 96] to produce variations of existing motion. Statistical modelling [24, 22, 132, 84, 100] is another popular method which learns a model of motion from a database which can be used to synthesise motion sequences based on their statistical characteristics. Finally, example-based methods [120, 11, 123, 70, 69] retains the original data to use in synthesis, whereby various subsequences are attached resulting in novel sequences.

The methodology adopted in this thesis is a combination of statistical modelling and example-based methods. Its application however, is extended to multiple data formats with multimodal interactive controllability, as well as other categories of formalisation that allows for robust and efficient real-time implementation. These are discussed in the later chapters.

The following sections review the motion editing techniques in more detail.

2.1.1 Constraint-based Methods

A fundamental issue with editing motion data is there is no prior knowledge of the motion, hence, no guarantee the resulting movement is physically correct. When animating from complex human pose configurations, this can lead to unnatural looking animations. To address this issue, Witkin et al. [136] introduced the *spacetime constraint*, which treats motion editing as a numerical optimisation problem. An objective function chooses a new motion that minimises the distance to the original, whilst adhering to all constraints.

This approach was then tailored to motion captured data, used in creating *motion transformations* [99]. By using a *character simplification* model, spacetime optimisation is used to fit the model to captured motion, essentially transforming the motion

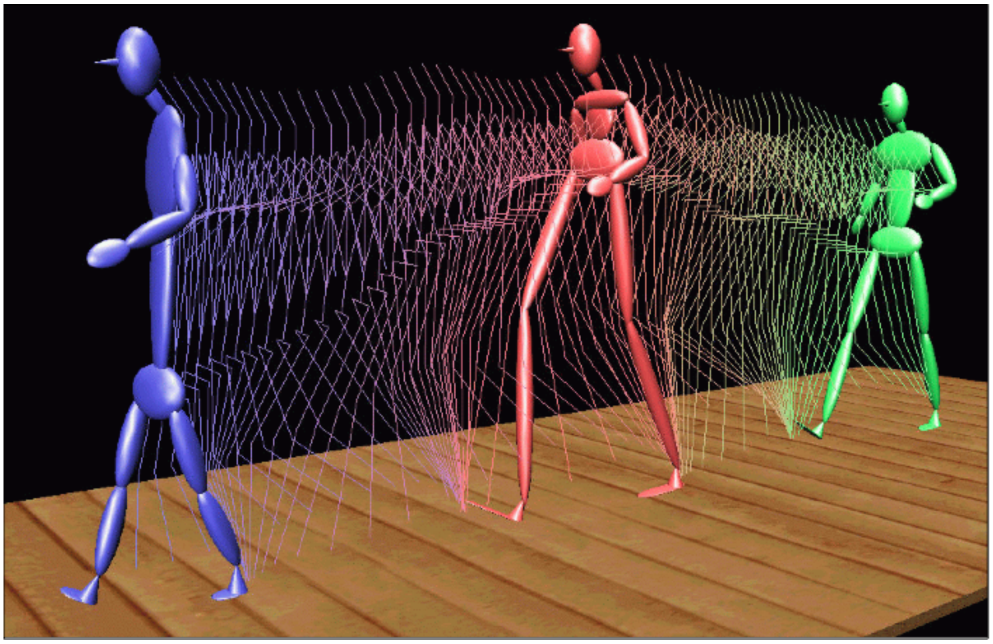


Figure 2.1: **Character morphing:** Motion captured character morphing [75]. The character is depicted at the modified frame.

by restricting its range of movement. The user can then edit the spacetime motion parameters, which are then used to reconstruct the final desired animation.

A similar optimisation technique is applied in [49, 75], whereby *retargetting* is used to map motion created for one character to another character of a different size. An example of *retargetting* is shown in Figure 2.1 where the user interactively morphs the size of the character whilst the original motion style is adapted to the character's altered proportions.

Since the optimisation is solved simultaneously for the entire animation sequence (as opposed to the individual frame), this results in high computational complexity and low user interactivity. To improve performance, Gleicher [48] suggests a trade-off between animation quality and computation complexity by ignoring some specified constraints. To enhance user interactivity, Cohen [32] proposes *spacetime windows*, whereby optimisation solutions are derived for subsequences of the animation as opposed to the entire animation.

Liu et al. [76], presents a method for synthesising complex dynamic motion from a simple animation. Input motion is analysed using a *constraint detection* method, which automatically determines linear and angular constraints. They demonstrate that such small sets of key parameters can be used to create realistic animations. However, this approach is best suited to highly dynamic motion and would otherwise fail when applied to low-energy movements such as walking.

These approaches tend to be computationally expensive and require a priori specification of constraints. As such, constraint techniques are best suited to off-line and post-processing applications, as opposed to real-time interactive animations.

2.1.2 Signal Processing

Since all joints in the human body are correlated, one can consider human locomotion as a time-vary signal of linked coordinates. As a result, signal processing techniques can be applied in editing a continuous stream of motion, and for blending between different types of human articulated movements. Witkin and Popovic [137] present *motion warping* which uses curve fitting to convert motion data into parameter curves. An animator edits key-frames on this motion curve and uses them as constraints by which a smooth deformation is applied, satisfying the key-frame constraints while preserving the realism of the original motion.

Bruderlin and Williams [26] treat motion parameters as sampled signals. These sampled signals can come from *spline curves* in keyframing systems, or from *tracked markers* in motion captured systems. Using recursive *multiresolution filtering*, they efficiently reduce the sample into multiple band levels. *Multitarget interpolation* is then used to blend the frequency bands between different articulated movements resulting in novel animations. Figure 2.2 illustrates the process of *Multitarget interpolating* between different frequency bands to create new animations.

Sudarsky and House [116, 117] propose the use of *non-uniform B-splines* for articulated human motion representation. Using set primitive operators on these motion curves, smooth manipulation of motion captured data is possible. *Non-uniform B-splines* provide more flexibility during curve fitting, but can lead to violations of joint limits.

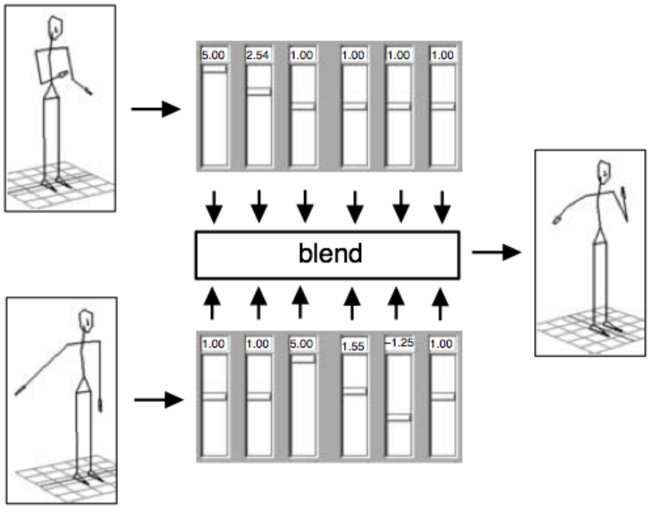


Figure 2.2: **Multitarget interpolation:** Multitarget interpolation between frequency bands [26].

Unuma et al. [126, 125] created a functional model of motion using *fourier series expansion.* The model is used to make variations of human behaviour via the interpolation and extrapolation of their fourier coefficients. Using a specified weighting value, the interpolation process can blend between different walks whereas the extrapolation process can exaggerate different walking styles. By considering the difference in fourier coefficients between two different data sets, they are also able to extract the combined fourier characteristics and model a novel walking style as a fourier characteristic function. This can then be blended with other motions, further extending the possible variations of human behaviour.

Utilising *wavelet analysis* [118], unlike fourier transforms, encompasses both frequency and time domain information, hence, any variation made will affect the whole motion resulting in smoother blends and better looking animations. Ahmed et al. [3] uses *wavelet transforms* to decompose motion curves into multi-resolution levels, whereby the low frequency resolution levels represent the main motion, and the high frequency resolution levels represent style and personality of the motion. Blending is made possible by linearly interpolating the coefficients of each resolution level independently, retaining more of the motion's natural variation.

Troje [124] uses *sine functions* to model walks. Using Principal Component Analysis (PCA), they extract relevant information from the data, representing them as discrete components. The temporal behaviour of these components are developed using a sine function, and sinusoidal curve fitting is used to parameterise them based on their respective frequency, amplitude and phase. This approach is able to capture the motion style inherent in the data and allows for simple blending of the sine coefficients of different motion styles using linear interpolation. A limitation to this approach is that it produces identical motion cycles which is not natural. No one walk cycle is identical to another, and a more realistic movement would present slight variations in every cycle.

A possible solution was presented by Bodenheimer et al. [21], suggesting adding noise to create the illusion of natural looking variability in cyclic human motion. The noise changes the joint angle trajectories over time whilst maintaining the characters walking style. To achieve this, the noise is added to the arm degrees of freedom (DOF) which imparts movement to the rest of the body. This is accomplished using a white noise process, whereby a continuous noise function produces a maximum amplitude at the extrema of a DOF during a walking cycle. As a result, the noise produced to the DOF of the joints are in phase with the body's movement resulting in natural variation during a walk cycle whilst maintaining the walker's style.

Motion variations created using signal processing methods, impact the entire motion clip resulting in smoother and better looking animations. However, these methods require the example motion to be time-warped to determine sequential correspondence between each component of each motion. Although this may be straightforward for periodic motions like walking and running, this can be difficult when dealing with sets of more aperiodic movements.

2.1.3 Multi-target Interpolation and Blending

Developing character animation using interpolation is one of the earliest techniques used in computer animation and is still used by many animators today. Traditionally used for generating intermediate frames from manually defined key-frames, a more recent functionality is multi-target interpolation of discrete points in a parametric space.

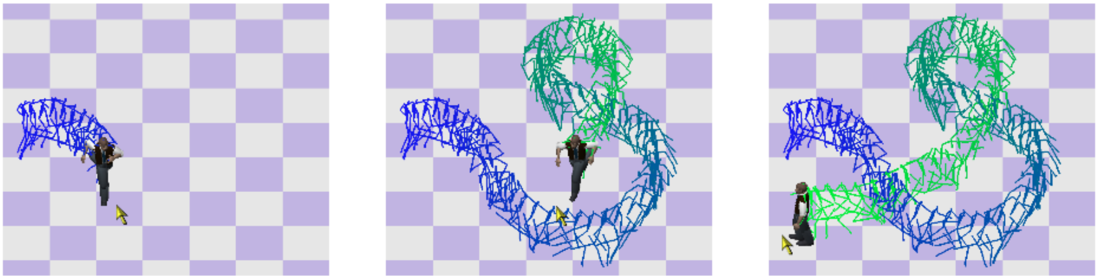


Figure 2.3: **Animated character control:** Locomotion chasing the mouse pointer [96].

Novel motion blends are generated by interpolating between multiple clips of time-varying correspondence, which can be used to lengthen animations, and create motion in-between sets to build up a parameterised space of movements.

Rose et al. [102, 103, 112] defined each animated pose as a hierarchy of rigid links connected to joints, whereby each joint contains one or more DOF. Each DOF's movement through time is represented as a uniform cubic B-spline curve. With this representation, they use *radial basis functions* (RBF) to interpolate between similar motion sequences in this space. Park et al. [96] extends this approach by using a *multidimensional scattered data interpolation* technique [113], incorporating *cardinal basis functions* instead of *radial basis functions*, which provides more efficient interpolation. Their parameter space is expanded to also incorporate speed, turning angle, and style (walk, run), which a user can interactively tune on-the-fly. This is demonstrated in Figure 2.3, where the locomotion of an animated character is guided by a mouse cursor in real-time.

Both of their approaches are able to synthesis motion based on groups of similar locomotion. However, it is only feasible for small databases since these groups of example motion are manually constructed with heavy constraints (key-event time). It would be a time consuming and tedious process when dealing with a large database. Since time-warping is required to derive frame correspondences, their approaches are not suited to acyclic motion.

Wiley and Hahn [135] combine sets of similar motions to create a multidimensional space of possible motions. The user specifies a pose and the system finds the subset of motions that are similar and occupy the desired parameter space. A *binary tree* *progression* of interpolation is used on each dimension to derive the complete position and orientation components of the pose. They use linear interpolation for each vector position component of the pose and spherical linear interpolation for each quaternion orientation component. The draw back to this technique is it does not guarantee the synchronisation of key-events which can result in reduced realism.

Ashraf and Wong [12] extend the application of *framespace interpolation* [53] to both cyclic and acyclic human motion which can come from multiple sources (as opposed to a maximum of 4). An interactive *forward kinematic interpolation* technique automatically determines correspondences in the source motion. Transition curve and inverse kinematic constraints are then used to guide smooth motion blending. However, their approach decouples the upper and lower half of the body, which produces a phase difference that needs correcting.

Cooper et al. [33] presented a method for interpolating motion by building a real-time motion controller. Each motion task (e.g. catching a ball) is parameterised by a control vector in a continuous space. Blendable motion clips associated to the same task are grouped together in clusters. Regions in this space that can not be blended are identified with an active learning system based on a set of error matrices. The system automatically generates what it believes to be an improved motion to occupy this space. If it is approved by the human-user, the system updates the motion controller with this motion, otherwise the human-user performs the required motion. This results in a continuous blend map that can smoothly blend all possible states to tasks in the controller.

Torresani et al. [122] use a theory of movement observations known as *Laban Movement Analysis* (LMA) to describe movement styles as points in a multi-dimensional space. Unlike the methods listed above, their system does not learn a parametric function of the motion, but instead a parametric function of how the interpolation and extrapolation weights applied to the data snippets relate to the style of the output sequences. This technique is developed from the combination of motion capture data snippets and learning parametric models of motion. A human expert is required to label the motion sequences according to the LMA-Effort factor of flow, weight and time. A non-linear regression model is fitted to these LMA labels and interpolation parameters, modelling style as a point in a 3D perceptual space and mapping the space of the motion style to the animation system parameters. However, the success of this technique is dependent on the human expert perception of style which may result in some inaccuracy if different human expert label the motion sequences.

Since multi-target interpolation methods create novel motion by blending between different samples of discrete frames, similar to the signal processing approaches, a time-warping phase is required to derive correspondences. Additionally, in many cases, extensive user intervention is necessary to accurately determine classes of similar motion types. Kovar et al. [68] introduce *registration curves* which uses a coordinate invariant distance function to automatically determine the relationships between frames based on the timing, local coordinate frame, and constraint state. However, their approach still relies on manually prepared motion samples with pre-determined constraints.

2.1.4 Statistical Model

A number of researchers have used statistical models to learn generalised motion characteristics for the synthesis of novel motion. Johnson et al. [62] uses a statistical model based on image observations [63] to learn simple human interaction. A stochastic tracking algorithm [60] is used to extract silhouettes of two individuals shaking hands. A *probability density function* (PDF) is learnt over a distribution of prototype vectors, derived by computing vector quantisation on the data. Combining the model with a markov-chain, human interaction (shaking hands) with a virtual human (2D tracked points) is possible. Galata et al. [45] extended this approach, using *variable length markov models* to encode high order temporal dependencies more easily. The interaction demonstrated in their work is quite simple (human user extends hand, virtual human extends hand), and although results are demonstrated on a video sequence, generation does not appear to be real-time and interactive.

Brand and Hertzmann [24] introduce *style machines*, which uses a statistical model to generate new motion sequences in a broad range of styles. They use *hidden markov models* (HMM) along with entropy minimisation procedures to learn and synthesise motion with particular styles. Their approach is similar to that presented in this thesis in that a gaussian process is combined with a markov chain. However, the work in this thesis also incorporates a projection mapping method for multimodal interaction control, which extends user controllability. Their work is also tailored to MoCap data and was not tested on temporal textures.

Bowden [22] developed a statistical model of motion known as a *Point Distribution Model* (PDM) by augmenting the discrete representation of PCA shapes with a markov chain. He uses PCA to perform eigenvector decomposition on the covariance matrix, then projects the data into a linear subspace with minimum lose of information. A fuzzy k-mean algorithm is used to segregate each data set into clusters. Each cluster corresponds to a state in a markov chain, and a first order markovian process is used to progress through the states. Linear interpolation is then used to further refine motion between fragments.

Carvalho et al. [28] also uses a PCA representation to train a motion model from motion captured data. However, they use a *prioritised inverse kinematics* strategy to apply motion constraints with different levels of importance. As opposed to using PCA, Grochow et al. [52] learns a PDF over character poses represented by a *scaled gaussian process latent variable model*. This model represents the data in a low dimensional latent space, and motion synthesis occurs by optimising the likelihood of new poses given the original poses.

Mukai and Kuriyama [84] present a technique of geostatistics called *universal kriging* that predicts continuous distribution of interpolation variables from samples. By considering all motion clips as spatial samples distributed in a multidimensional space, the correlation between samples is used to estimate a statistical model known as a *variogram function*. Univeral kriging predicts a distribution of the variogram function by statistically estimating the correlations between the dissimilarity of motions and distance in the parameteric space.

Wang et al. [132] proposed a non-parametric dynamical system based on a *gaussian processes latent variable model*, which learns a representation for a nonlinear system. Their approach models pose and motion separately using a dynamic process and obser-

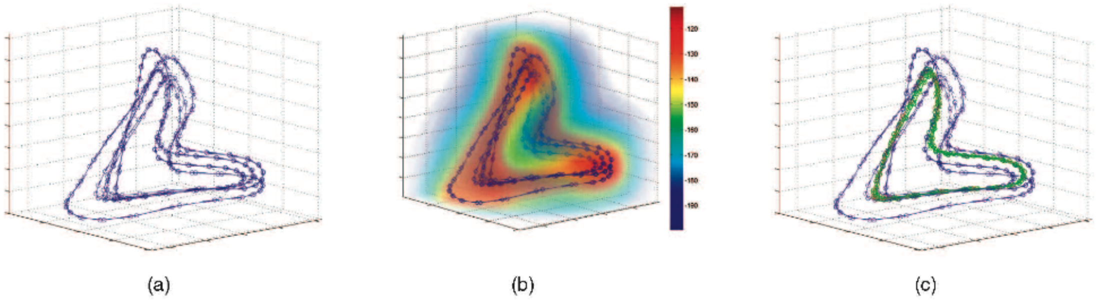


Figure 2.4: **Latent variable model:** Latent variable model, learnt for walking sequences from three different subjects. (a) Learnt latent coordinates, (b) variance plot, (c) Green lines shows the dynamic predictive distribution [132].

vation process respectively. This high dimensional data is efficiently reduced to a low dimensional latent space, resulting in a non-parametric system that can account for uncertainties in the model. This is illustrated in Figure 2.4, where 2.4(a) is the learnt latent coordinates, 2.4(b) is the variance plot, and 2.4(c) shows the dynamic predictive distribution in green.

Pullen and Bregler [100] introduce the idea of synthesising motion by extracting the *motion texture* i.e. the personality and realism, from the motion and using it to drive hard constraints such as foot positions on the floor. They comprise their motion of three important features, frequency band, phase, and correlation. These features are represented with a *kernel-based probability distribution*. This distribution is used to synthesise the walk and a gradient based method is used to optimise the data. Part of this approach is adopted in this work whereby a *multivariate probability distribution* is used to model the data and synthesise a walk. However, blending between different distributions is also made possible, as well additional formalisations to facility real-time animation and interactive control. In the subsequent chapter, this approach is further extended to an exemplar-based method, using the probability distribution to determine the most like start and end pose configuration given a user specified action.

The statistical approaches mentioned above work well in generalising motion characteristics, however, they are all tailored to animate only 2D and 3D marker points. Basharat and Shah [15] use *chaos theory* to learn a generalised model for nonlinear dynamic systems, which can synthesise both motion capture data and temporal textures. By representing motion samples as a chaotic system, kernel regression is used to predict future points from an initial configuration. Their approach learns a generalisation for an individual motion type which is best suited for extending motion samples but not for motion blending. Also, their system does not provide real-time interactive control.

Chapter 3 presents a novel approach to creating a statistical model for generating real-time animation. This approach can animate both marker points and temporal textures in video, and allows real-time interactive control of animations. The user can choose to synthesise any motion type inherent in the data, and interactively blending between them.

2.1.5 Example-based Methods

Motion synthesis using example-based methods, i.e. retaining the original data to use in synthesis, provides a more attractive alternative to statistical models since there is no loss of motion detail.

Molina-Tanco and Hilton [120] adopts this approach, proposing a multi-level statistical model for motion capture data. The first level of the model consists of a markov chain of the joint trajectories which allows the generation of motion by traversing states. By dividing the joint space into clusters using a k-mean classifier, each markov state corresponds to a region or cluster. On its own, this level does not produce high quality motion due to the compression performed by the statistical model. However, in the second level, using bayes theorem, they relate the markov states with segments of the original motion in the database, allowing the generation of realistic motion based on the segments.

Arikan et al. [11] developed a system that allows the user to synthesise motion by creating a timeline with annotated instructions such as *walk*, *run* or *jump*. The system then assembles frames collected from a motion database allowing the final motion to perform the specified actions at specific times. The user may also specify constraints, requiring the motion to perform a particular pose, or move to a particular position and orientation at a given time. This is achieved using *Support Vector Machine* (SVM) classifiers which generalise the user annotations to the entire database. The synthesis algorithm is based on successive dynamic programming optimisation. It finds blocks of motions that can fit together in a motion sequence and at the same time satisfy the annotations and other low level constraints. The entire process is interactive and the user can change desired motion properties on-the-fly. For this approach to be successful, a relatively large database is needed to work with their optimisation technique otherwise it may result in highly repetitive motion generation.

Treuille et al. [123] developed a system that synthesises kinematic controllers which blend subsequences of precaptured motion clips to achieve a desired animation in real-time. Using a parametric *value function*, their controller selects sequences of clips to achieve a user specified objective. Objectives can include navigation as well as and obstacle avoidance. The limitation to this approach is it requires manual segmentation of motion subsequences to a rigid design in order to define appropriate transition points.

Representing motion transitions using a *motion graph* [101, 111, 13, 17, 50], originally introduced by Kovar et al. [70], provides additional control. It uses sequences from the original data and automatically generates transitions to perform an optimal graph walk that satisfies user-defined constraints. Each motion is defined by the position of its root joint and the quaternion representation of each joint. They use similarity matrics to determine fragments of similar motion. By representing frames in terms of point clouds, they calculate the weighted sum of squared distance between corresponding points in the clouds. If the distance is below a user-specified threshold, the relative motions are considered similar. These groups of similar motions can then be blended by linearly interpolating the corresponding root positions and using spherical linear interpolation of corresponding joint rotations. Figure 2.5 shows an example of synthesising human articulated motion using a motion graph. The curve represents the path the character is required to walk, and the transition points are indicated where the curve changes colour.

Kovar and Gleicher [69] improve on this further by analysing the correspondences themselves, assuming increased simplicity in finding corresponding frames when amongst similar motion types. This addition increases the robustness of the search for simi-

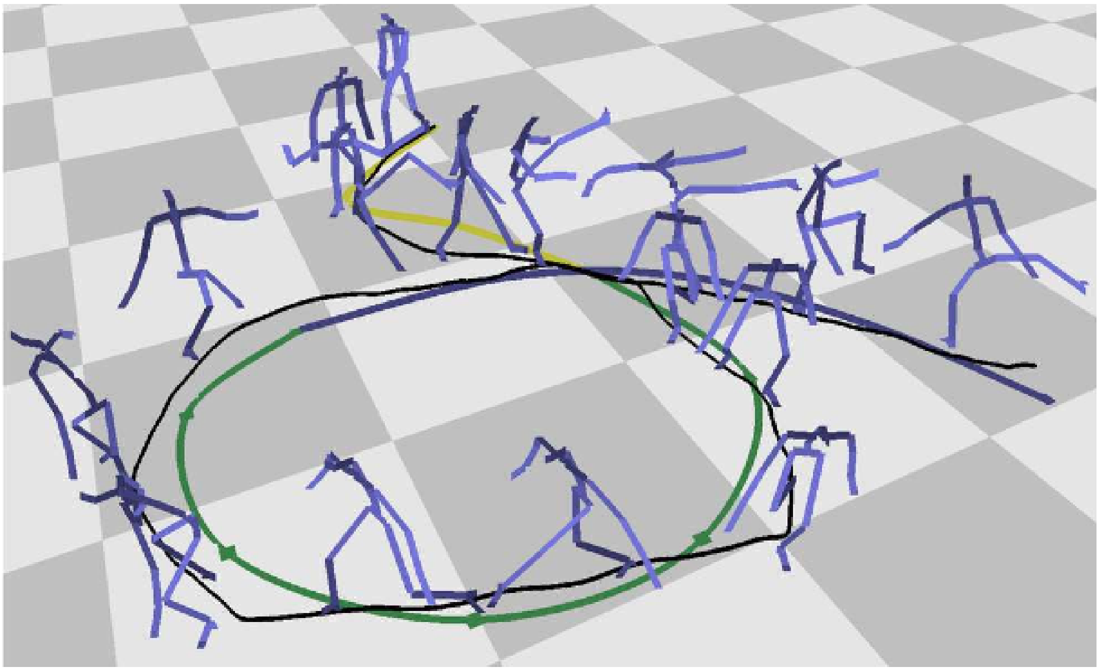


Figure 2.5: **Motion graph:** Example of motion synthesis using motion graph [70]. The curve represents the user desired path, and the curve's colour indicates a transition to another motion type.

lar motion fragments. They also include a blending weight constraints that limits the amount of allowable extrapolation, projecting unattainable motion requests back onto the accessible portion of the parameter space.

The example-based model presented in this thesis extends on motion graph by using a pose space PDF to derive the likelihood of a pose given the data, ensuring better quality transitions. The work within this thesis also presents a more efficient approach to deriving transition points based on k-medoids, making this approach more appropriate for large data sets of video.

Lee et al. [74] used interactive controllers to animate an avatar from human motion captured data. They present three control interfaces: selecting a path from available choices to control the motion of the avatar, manually sketching a path (analogous to *Motion Graphs* [70]), and acting out motion in front of a camera for the avatar to perform. The motion controller in Chapter 4 is multimodal and demonstrated driven by keyboard, mouse, and gesture interfaces. The controller is also extend to the audio domain, using audio MFCC features to drive the motion model. In Chapter 6, the controller is further extended to the social domain, using mined social signals to derive animations. Previous approaches to modelling motion driven by audio features, have been used for lip-syncing a facial model [25, 23], or animating the hand and body gestures of a virtual avatar [115]. In these examples, audio signals are used to animate a speaker or performer. Jebara and Pentland [61] touched on modelling conversational cues and proposed Action Reaction Learning (ARL), a system that generates an animation of appropriate hand and head pose in response to a user's hand and head movements in real-time. However, this does not incorporate audio.

2.2 Temporal Texture Synthesis

Early approaches to texture synthesis were based on parametric [57, 35] and non-parametric [41, 134, 94] methods, which create novel textures from example inputs. Approaches to static texture synthesis paved the way for temporal texture synthesis methods, often used in the movie and gaming industries for animating photo-realistic characters and editing video scenery.

There are two common placed methods to temporal texture synthesis; *generative methods* and *example-based methods*, both of which have been adopted in this thesis. Similar to the motion capture approaches, generative methods [119, 14, 133, 18, 38] learn a generalisation of time-vary relationships in texture to generate novel video sequences. This method is best suited to stochastic textures such as waterfalls, flames etc, since the inconsistencies in the generalisation are more evident when attempted on precise temporal textures such as the human facial expressions and gestures. However, example-based methods [25, 23, 71, 44, 106], retain the original texture data to use in synthesis, making it more appropriate for synthesising precise temporal textures. However, an intuitive approach is needed to derive appropriate transition points, to disguise the switching/blending between discrete video subsequences.

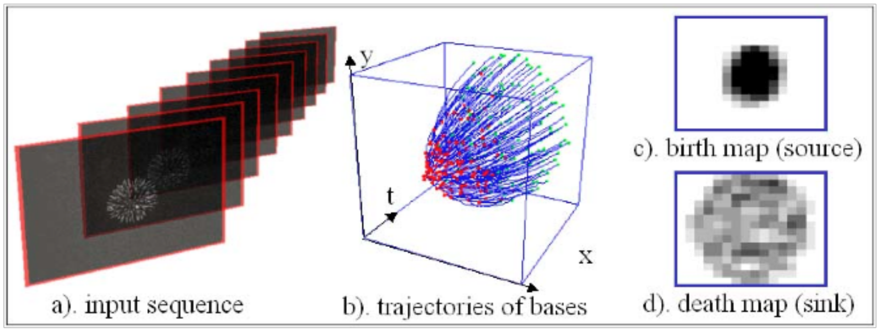


Figure 2.6: **Computed trajectories of movetons:** The computed trajectories of fireworks and the source and sink maps [133].

2.2.1 Generative Methods

An earlier approach to temporal texture synthesis was introduced by Szummer et al. [119], who used a *spatio-temporal autoregression* model (STAR) [31] to model image sequences. By representing each pixel as a linear combinations of surrounding pixels, a neighbourhood structure of the model is defined. Using a least square method, estimating unknown pixel parameters is made possible.

Bar-Joseph et al. [14] use *wavelets* to generalise temporal textures into signals. By learning a hierarchical multi-scale transform of the signal, conditional probabilities are used to traverse paths, resulting in new random textures.

Wang and Zhu [133] combines algorithms from both texture and motion analysis (motivated by vision and graphics methods) to create a generative model for temporal texture synthesis. They represent temporal textures as superposition of linear bases. *Movetons*, which are moving elements in the texture, are clustered into groups of spatial adjacent bases. A markov chain is used to model dynamics, and the source and sink movetons are modelled by birth and death maps. Figure 2.6 shows an illustration of their approach. By editing the birth map, they are able to synthesise more fireworks. This approach however, is quite computationally expensive and can only provide near real-time animations.

Similarly, instead of *Movetons*, Bhat et al. [18] allows a user to synthesise specific texture dynamics in a video using *texture particles*. The user can define flow-lines in

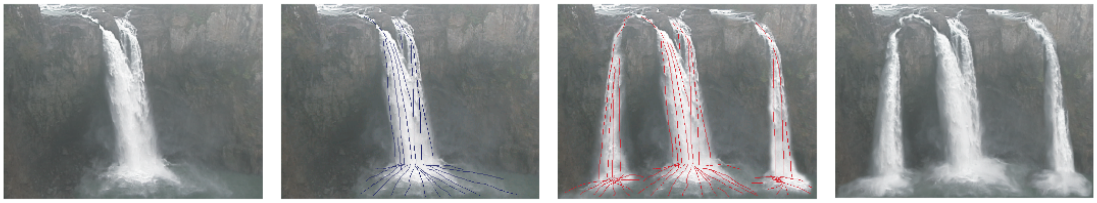


Figure 2.7: **Texture particles:** Synthesising new video by manipulating flow-lines [18].

the sample video where dynamics and texture variations are captured. Synthesis is as a result of blending the textures along these particles over time. This approach differs from [133], in that the user can intuitively edit a video scene by sketching flow-lines on top of an image. This is demonstrated in Figure 2.7 where given a video sequence of a waterfall, additional waterfalls with appropriate texture dynamics can be added to the scene.

Doretto et al. [39, 114, 38] present an approach for modifying temporal behaviour of dynamic textures. *Linear Gaussian models* are used to synthesise novel sequences with the same characteristics as the original video. By using eigen-decomposition to extract eignvalues of the sample textures, and representing the eignenvalues in polar coordinates, speed manipulation is made possible by altering their normalised frequencies.

Although these methods work well in generalising temporal texture for synthesis, they do not provide real-time interactive control of the objects undergoing motion. They are also tailored to temporal textures and cannot be applied to other motion formats. The generative method presented in the thesis, can synthesise discrete states of different motion types inherent in the data, as well as allowing interactive real-time control for transitioning to different movements. The approach presented in this thesis is also applicable to both temporal textures and motion capture data.

2.2.2 Example-based Methods

Bregler [25] was one of the first in adopting example-based methods by introducing *video rewrite*. Video rewrite lip-syncs a facial model. Using sample videos of a person speaking naturally, new video can be created of the same person mouthing words they

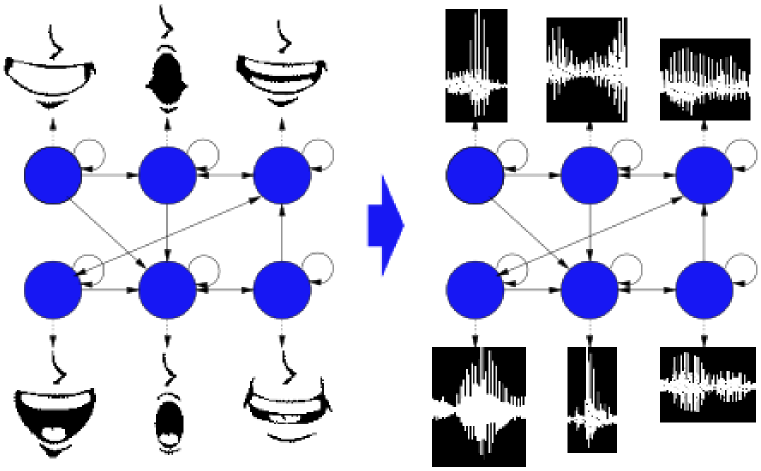


Figure 2.8: **Voice Puppetry:** Reuse of the facial HMM's internal state machine in constructing the vocal HMM [23].

had not spoken. Following a similar approach to concatenative speech synthesis in [83], this is achieved by attaching together visemes of the mouth region extracted from the original data to match the new utterance. These visemes are blended back onto the face resulting in the final animation.

Brand [23] expands on this, introducing *Voice Puppetry*, which also incorporates dynamic information of the entire face and not just the lip region. Figure 2.8 illustrates his approach. Given an input video, the face alone (not including audio) is analysed using entropy estimation. This is used to learn facial dynamics and build a facial HMM. An occupancy matrix is used to associate the synchronised audio (voice) to each facial state resulting in an audio driven vocal HMM. Given a new vocal signal, a Viterbi algorithm is applied to the vocal HMM to animate the optimal sequences of facial configurations.

In both these examples, audio signals are used to animate a speaker or performer. In this thesis, the example-based approach uses audio signals to animate a listener in a conversation. The audio-video mapping is not based on specific phonemes but rather non-verbal inclinations to trigger backchannel responses from the listener.

Another well know method was introduced by Kwatra et al. [71], who generated perceptually similar patterns from a small training data set, using a *graphcut* technique based on Markov Random Fields (MRF). Combined with an approximative offset search techniques, graphcut automatically determines optimal patch regions from a sample, which can then be copied to an output to generate a new and larger output. This method is applicable to static and temporal textures, however only suited to stochastic sequences.

In some cases, example-based techniques used for the synthesis of motion captured data are similar to example-based techniques used for temporal texture synthesis of videos. By substituting pixel intensities (or other texture features) with marker co-ordinates, and applying motion constraints suited to the desired output, a similar framework can be extended to both domains.

Schödl et al. [106, 105] introduce *Video Textures* which compute the distances between frames to derive appropriate transition points to generate a continuous stream of video images from a small amount of training video. Their input data is represented as a Markov process which is used to determine the likelihood of transitioning between discrete frames at ends of different subsequences. To account for mirroring effects during transitions, appropriate transition points are chosen to have both frame-to-frame similarity and temporal similarity over a frame window. Their system was demonstrated on several examples including a random play video of a human face, and a mouse controlled fish, whereby a mouse cursor was used to guide the path of the fish with different velocities.

Similarly, Flagg et al. [44] presents *Human Video Textures* where, given a video of an actor performing various actions, they produce a photo-realistic avatar which can be controlled, akin to a game character. Their data set consists of both recorded video (using a single high definition camera), and 3D motion captured markers placed on the performer. They use the motion captured data to identify transition points in the video clip. A translation alignment of the marker points is done first before a similarity measure. To generate transitions, they use a 15 frame window consisting of 3 phases; pre-transition, transition, and post-transition. Each phase consists of 5 frames. There is no blending in the pre- and post-transition phases, only moving least square (MLS) warping (used for alignment of corresponding frames). This is demonstrated in Figure

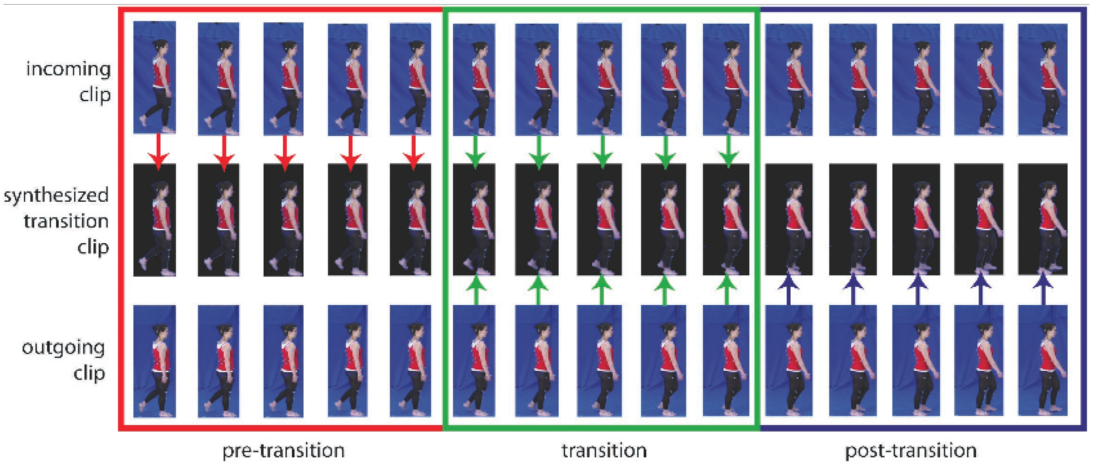


Figure 2.9: **Human video textures:** Transition process involving interpolating corresponding clips [44].

2.9. In the transition phase, linear interpolation of the marker points (in 2D) are used for blending. To account for self-occlusion of the arm during a walk cycle, they segment the arm from the main body using MRF and compute separate warps and blending for them.

Work in this thesis is similar to both these cases, whereby temporal texture sequences are segmented based on distance/similarity, and used to create novel sequences. However, in these cases, human texture synthesis is performed on periodic data, or on data constrained to guarantee the actor returned to a neutral pose. In this thesis, the example-based method performs texture synthesis on natural human conversation data, whereby human social behaviour is neither periodic nor predictable. A novel combination of social behaviour can be generated by a user, which is extended to an autonomous social interactive system, using conditional probabilities derived by a social dynamics model.

Using emotions to interact with video animations was also suggested in [59], where the emotional expressions of an audience evokes *emotional contagion* [55] of facial video portraits. However, this differs from the approach presented in this thesis in that, specific mined rules govern the social responses as opposed to synchronised mimicry.

2.3 Social Behaviour Analysis

Traditional social interaction research can be grouped into two main categories: emotion based on cognitive psychology [42], and linguistics based on dialogue understanding [9, 65]. Although emotion understanding is of vital importance in how people socially interact, emotion recognition in a natural conversation is a very complex problem and would require extensive data and research in deducing social trends. Also, structured dialogue can not be easily interpreted to observe generalised social behaviour.

Other methods utilise machine learning models such as HMM [46] and Dynamic Bayesian Networks [36], and apply it to generic features in audio signals and pixel intensities to discern social behaviour. Schuller et al. [109, 110] uses a fusion of audiovisual features such as, *Active-Appearance-Model-based* facial expressions, and linguistic analysis, combined with Support Vector Machines to perform classification and regression of interest levels.

Bianchi-Berthouze et al. [66, 20, 19], focus primarily on using body posture for emotion recognition. With 55% of non-verbal communication expressed primarily through body language [78], they demonstrate accurate recognition through the use of body language alone (i.e. without facial features or audio signals), even across different cultures.

Work on social analysis in this thesis differs from those listed above in that, association rule mining is applied to audiovisual non-verbal social signals to discern social behaviour and deduce specific rules that present prominent trends.

2.3.1 Non-Verbal Social Signals

Psychological studies have proven that observing non-linguistic/non-verbal, unconscious social signals [5], can provide effective information in social interaction understanding [67]. There are five main groups of non-verbal behavioural cues [56]: *physical appearance*, *gestures and postures*, *face and eye behaviour*, *vocal behaviour*, *space and environment*. The most common approach in capturing these social signals are by using capture devices like microphones and cameras, although more elaborate methods exist such as smart meeting rooms [131] and mobile wearable devices [40].

A number of researchers have used machine analysis of non-verbal social signals to interpret social behaviour. Aran et al. [7] detect dominant people in a conversation using audio-visual cues applied to a rule-based estimator. The idea of *Social Signal Processing* [128, 129], originally introduced by Pentland [98] and adopted in this thesis, is to use visual and vocal analysis to understand social behaviour and predict outcomes of dyadic interactions to enable a *Human-Centred* computing paradigm. This is achieved using *textures* (i.e. speaker energy and amount of movement) [97] from multimodal social signals. Similarly, Curhan et al. [34] uses these *texture* features to predict outcomes of negotiations based on thin slices [6] of employment negotiation data. Although these methods perform well in predictions, they rely on psychological observations to derive prior assumptions of what is positive or negative social behaviour. This may not be accurate in all social contexts. Also, their approach is unable to discern co-occurence of social signals of multiple modes, as these more complex dependencies are difficult to identify. However, in this work, a *social dynamics model* is introduced, which utilises data mining to derive tangible rules for visualising multimodal social interaction and for accurately predicting social context. This is achievable independent of prior psychological evaluations, relying solely on the trends in the data.

Eagle et al. [40] introduce *reality mining* which use mobile devices, like smart badges and cellular phones, to extract proximity and vocal information to derive social networks. Their approach differs from that applied in this thesis in that, in this thesis, multimodal social signals are used as features for association rule mining, with the aim of deriving specific rules that govern conversation interest.

2.3.2 Socially Interactive Animation

With the means of understand social behaviour, this paves the way for improvements in human-computer interactive systems. Such systems include *conversational agents* such as avatars, which can both communicate and interpret information from a human user, akin to a one-to-one interaction between two people. Such systems bridge the gap between man and machine interfaces. With technological advancements, avatars can resemble humans, understand dialogue, and communicate verbally, however, they still lack the understanding of natural human non-verbal behaviour essential for human interaction. As such, a vast amount of research has been adopted to address this issue.

Maatman, Gratch and Marsella [77], create a computer generated character that responds non-verbally (as an artificial listener) to a speakers, based on their audio signal, posture and head orientation in real-time. Similarly, Schröder et al. [107] proposed SAL (Sensitive Artificial Listener), which engages the user in a conversation by observing the user's emotions and non-verbal expressions. Cassell et al. [29] propose *BodyChat*, which is a system that allows a user to communicate via text whilst their avatar generates non-verbal responses to accompany the user's input. The avatars can be controlled autonomously, manually, or both. Using an ANOVA and subsequent post-hoc t-test, their results showed that users found autonomous controlled avatars to be more natural, expressive, communicative. These approaches differ from work presented in this thesis in that, their animated character's social interaction are predetermined (knowledge driven) based on prior assumptions (i.e. mimicry), where as the interaction of the artificial listener presented in both Chapter 4 and 6 are data driven, and only based on interaction learned from a conversation data set.

Pelachaud et al. [87] propose a partially data driven approach for generating non-verbal social behaviour in virtual characters. Their model is based on both manually annotated video data and descriptions found in literature, to decode the collection of expressions that lead to certain human emotional behaviour. Gillies et al. [47] present a fully data driven approach, where a speaker's audio signal is used to drive the full body MoCap animation of a listener. This approach is similar to work in Chapter 4, where a motion model is combined with reinforced learning to animate appropriate responses of a listener based on a speaker's audio. Though they attempt to add context information to their synthetic listener, it's responses are only driven by speech and can only operate in the simplest of contexts, i.e. [*energetic voice* ⇒ *respond*], or [*calm voice* ⇒ *respond*]. In Chapter 6, our approach extends this idea by using data driven multimodal mined rules to generate social responses from an avatar. This approach uses both audio and visual social signals, and can operate in more diverse social contexts.

2.4 Summary

This chapter has presented a review of related research in motion capture synthesis, temporal texture synthesis, and social behaviour analysis.

Generative methods to both motion capture and temporal texture synthesis, can generalise motion characteristics, however, with the need for pre-defined constraints and heavy pre-processing, not all provide real-time interactive control, and are engineered for a single data format. Example-based methods for synthesis produces better looking animations, however, most other methods utilise a tedious approach in deriving transition points, best suited to smaller data sets. In the following chapter, these problems have been addressed. Both a generative and example-based motion model are presented, capable of modelling motion in various formats and providing an interactive user interface. An intuitive approach to deriving transition points is utilised, making this methods applicable to data sets of varying sizes.

A challenging and largely unaddressed problem is in generating autonomous socially interactive avatars. To achieve this, an understanding of natural human social behaviour is needed. As such, a review of popular methods to social behaviour analysis was presented. Although understanding human social behaviour using *texture* works well in predictions, it does not provide any information on specific exchanges in social signals. To address this, a social dynamics model is developed that provides tangible rules of specific exchanges in social signals. These rules are used as conditional probabilities to drive the motion model.

The next chapter presents a generative method for motion capture and temporal texture synthesis.

Chapter 3

Generative Model for Real-Time Motion Synthesis

This chapter presents a method for reusing motion capture (MoCap) and video data by learning a generative model of motion. The model allows real-time motion synthesis and blending whilst providing it with the style and realism present in the original data. Unlike other generative models of motion, this approach is applicable to both MoCap and temporal textures in video, and provides real-time interactive control of the generated animations. This approach also extends on work by Pullen and Bregler [100], allowing linear blending of PDFs and their constraints, in order to perform plausible transitions to different motion types.

Using *Principal Component Analysis* (PCA), the high dimensional data samples are projected into a lower dimensional eigenspace (Section 3.2). A statistical generalisation of the motion sequences are learnt using a *multivariate probability distribution* (Section 3.3.1). As such, two *Probability Density Functions* (PDFs) are derived. One PDF estimates the likelihood of pose/frame, and the other estimates the likelihood of motion from a given configuration. Combined with *Gaussian noise* (Section 3.3.6) for naturalistic motion variation, and a *gradient-based optimisation* (Section 3.3.4) to derive the optimal path, novel natural motion synthesis is made possible. To facilitate real-time animation, the PDF is combined with a *kd-tree* for fast Gaussian approximation (Section 3.3.2).

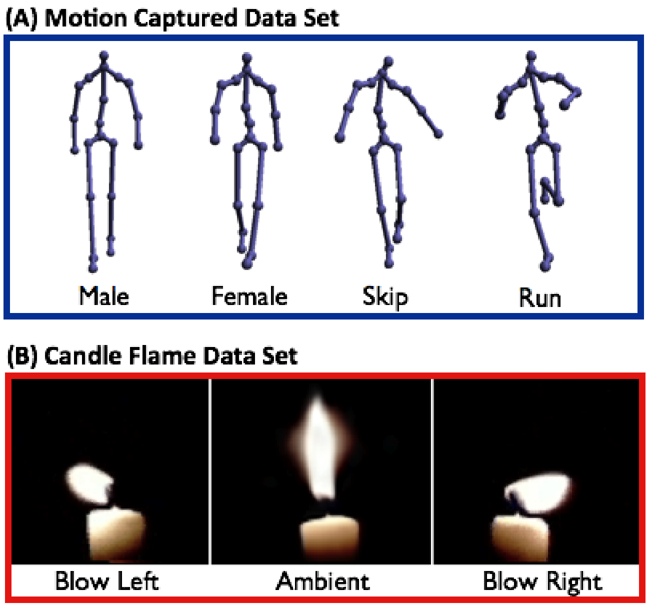


Figure 3.1: **Data set:** (A) Sample 3D motion captured data of different walks, (B) Sample candle flame video recording of a flame undergoing motion

3.1 Data Set for Generative Model

Given a motion sequence $\mathbf{X}$, each frame is represented as a vector $\mathbf{x}_i$ where $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{N_T}\}$ and N_T is the number of frames. Various motions can be modelled by the system. We demonstrate 3D motion, and rgb pixel intensities in two examples:

- **3D MoCap Data:** The motion captured data used in this work are in a format which details the 3D Cartesian coordinates (x,y,z) [30] for all the markers corresponding to the frames for the full body MoCap data, although similar approaches can be applied in polar spaces. Four different walks are used: *male walk*, *female walk*, *skipping*, and *running*. The user can synthesise a novel walk sequence in real-time and blend between different types of walks.

- **Candle Flame:** The movement of a candle flame is synthesised whereby the user has control over three discrete states: *ambient flame*, *flame blow left*, *flame blow right*, and can blend between them.

In each cases, each time step i of the data to be modelled is vectorised as $\mathbf{x}_i = (x_{i1}, y_{i1}, z_{i1}, ..., x_{ib}, y_{ib}, z_{ib}) \in \Re^{3b}$ for a 3D contour of b points and $\mathbf{x}_i = (r_{11}, g_{11}, b_{11}, ..., r_{\mathrm{xy}}, g_{\mathrm{xy}}, b_{\mathrm{xy}}) \in \Re^{\mathrm{xy}}$ for an x × y image.

3.2 Dimension Reduction and Eigenspace Projection

To reduce the complexity of building a generative model of motion, Principal Component Analysis (PCA) [4, 104] is used for dimensionality reduction. Since the dimensionality of the resulting space does not necessarily reflect the true dimensionality of the subspace the data occupies, only a subset of the eigenvectors are required to accurately model the motion.

For a given D-dimensional data set $\mathbf{X}$ as defined in Section 3.1, the D principal axes $\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_D$ are given by the D leading eigenvectors of the sample covariance matrix:

$$\mathbf{S} = \frac{1}{N_T} \sum_{i=1}^{N_T} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \tag{3.1}$$

where $\boldsymbol{\mu}$ is the sample mean $\boldsymbol{\mu} = \frac{1}{N_T} \sum_{i=1}^{N_T} \mathbf{x}_i$. An eigen decomposition gives $\mathbf{S} = \sum \boldsymbol{\lambda}_i \mathbf{T}_i$, $i \in \{1, ..., D\}$, where $\boldsymbol{\lambda}_i$ is the ith largest eigenvalue of $\mathbf{S}$.

The dimension of the feature space $|\mathbf{x}_i|$ can be reduced by projecting into the eigenspace

$$\mathbf{y}_i = \mathbf{V}^T(\mathbf{x}_i - \boldsymbol{\mu}) \tag{3.2}$$

where $\mathbf{V}$ is the projection onto the eigenspace $\mathbf{V} = [\mathbf{T}_1, ..., \mathbf{T}_d]$, $\mathbf{T}_i$ are the eigenvectors, $\boldsymbol{\lambda}_i$ the eigenvalues, $\boldsymbol{\mu}$ is the sample mean, and d is the chosen lower dimension $d \leq |\mathbf{x}_i|$ such that $\sum_{i=1}^{d} \frac{\lambda_i}{\Sigma \forall \lambda} \geq .95$ or 95% of the energy is retained. $\mathbf{Y}$ is defined as a set of all points in the dimensionally reduced data where $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_{N_T}\}$ and $\mathbf{y}_i \in \Re^d$. This results in a d-dimensional representation of each frame in the sequence. This representation reduces the computational and storage complexity of the data whilst still retaining the time varying relationships between each frame.

Figure 3.2 (A) shows a plot of the first mode against the second mode for all sequences, i.e. the data projected onto the 2 eigenvectors that correspond to the two largest eigenvalues. Their distributions produce a geometric shape characteristic of a cyclic motion, showing that the projection retains the non-linearity of cyclic movement. The sequences vary in number of frames but all contain at least one complete cycle of the motion to be modelled. The *male walk* sequence consists of 182 frames, the *female walk* 229 frames, the *skipping* 135 frames, the *running* 35 frames, and the *candle flame* 3000 frames. The number of frames were based on the available data and were not chosen apriori. This however highlights the diversity of the approach, which is capable of synthesis and blending regardless of data size.

3.3 Generative Model for Motion Synthesis using PDF

3.3.1 Training a PDF

A statistical model of the constraints and dynamics present within the data can be created using a PDF. A PDF of appearance is created using kernel estimation where each kernel $p(\mathbf{y}_i)$ is effectively a Gaussian centred on a data example $p(\mathbf{y}_i) = G(\mathbf{y}_i, \Sigma)$. Since we want our probability distribution to represent the dimensionally reduced data set $\mathbf{Y}$ as noted in Section 3.2, the likelihood of a pose in pose space is modelled as a mixture of Gaussians using multivariate normal distributions.

$$P(\mathbf{y}) = \frac{1}{N_T} \sum_{i=1}^{N_T} p(\mathbf{y}_i) \tag{3.3}$$

where the covariance of the Gaussian is:

$$\Sigma = \alpha \begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\lambda_d} \end{pmatrix} \tag{3.4}$$

Figure 3.2 (B) shows a plot of such a distribution for each data set with the first mode plotted against the second mode. The width of the Gaussian in the i^{th} dimension is set to $\alpha\sqrt{\boldsymbol{\lambda}_i}$.

If $\alpha = 0$, i.e. the variance is set to 0, the synthesis will not generalise and simply replay the original data. If α is too high, there is no constraint upon pose and the resulting animation will be destroyed. Figure 3.3 illustrates this point. It shows the

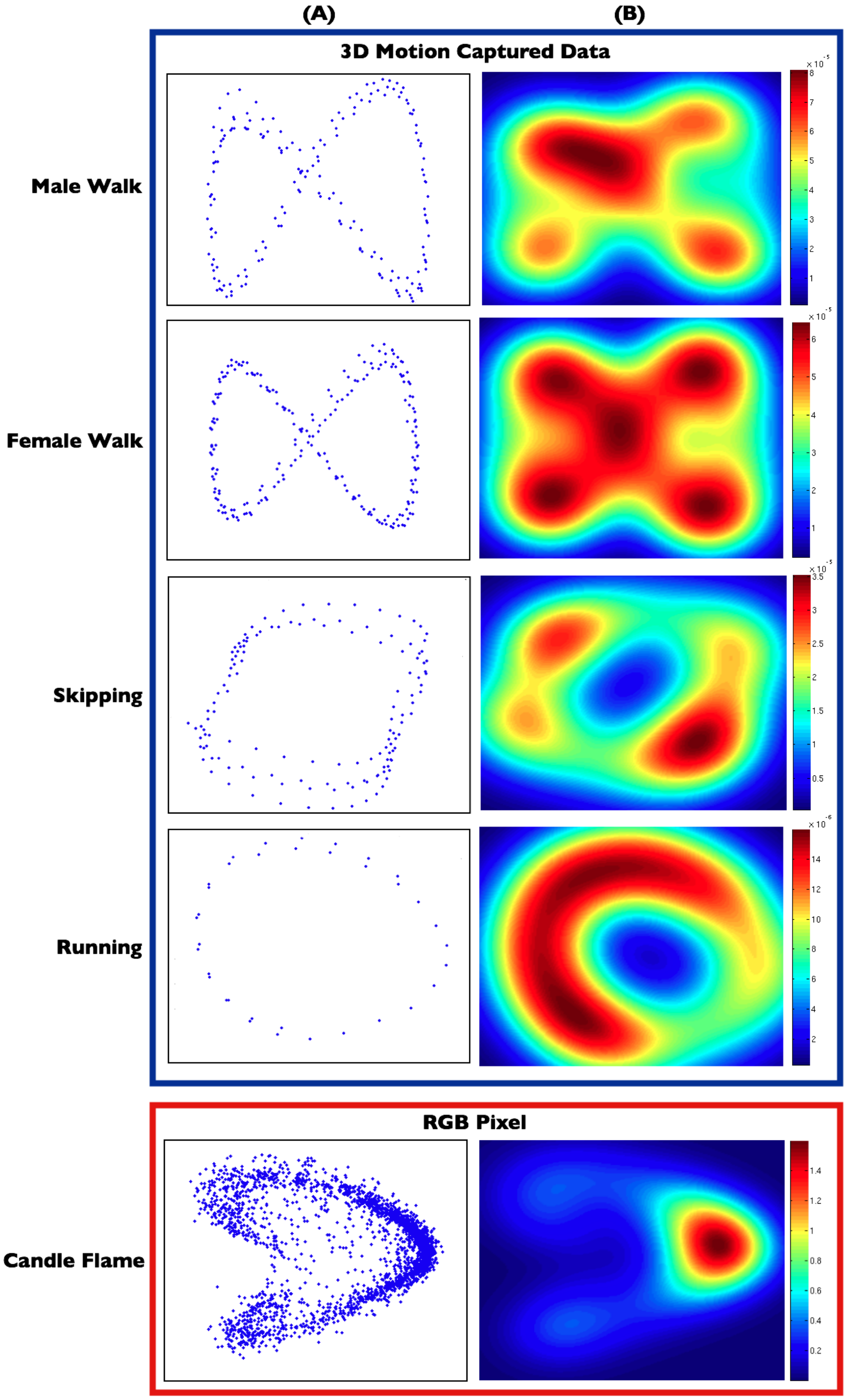


Figure 3.2: **PCA projections and PDFs:** (A) Plot of PCA projection of the first mode against the second mode. (B) Probability Density Function (PDF) for all sequences. The width of the kernel is $\alpha\sqrt{\lambda_i}$ where $\alpha = 0.25$.

PDF projection of the two leading eigenvectors for all data sets using different values of α. In column (A), α is too large with a value of 1, and in column (D), α is too small with a value of 0.025. However, the intermediate plots provides reasonable representation of the data. As the eigenvalues are based on the variance of the overall data set, this allows the PDF to scale appropriately to the data. Therefore, we chose α experimentally to provide a good trade off between accurate representation and generalisation, but it is important to note that this parameter remains fixed for all data sets. For all experiments $\alpha = 0.25$ (as shown in column (B)).

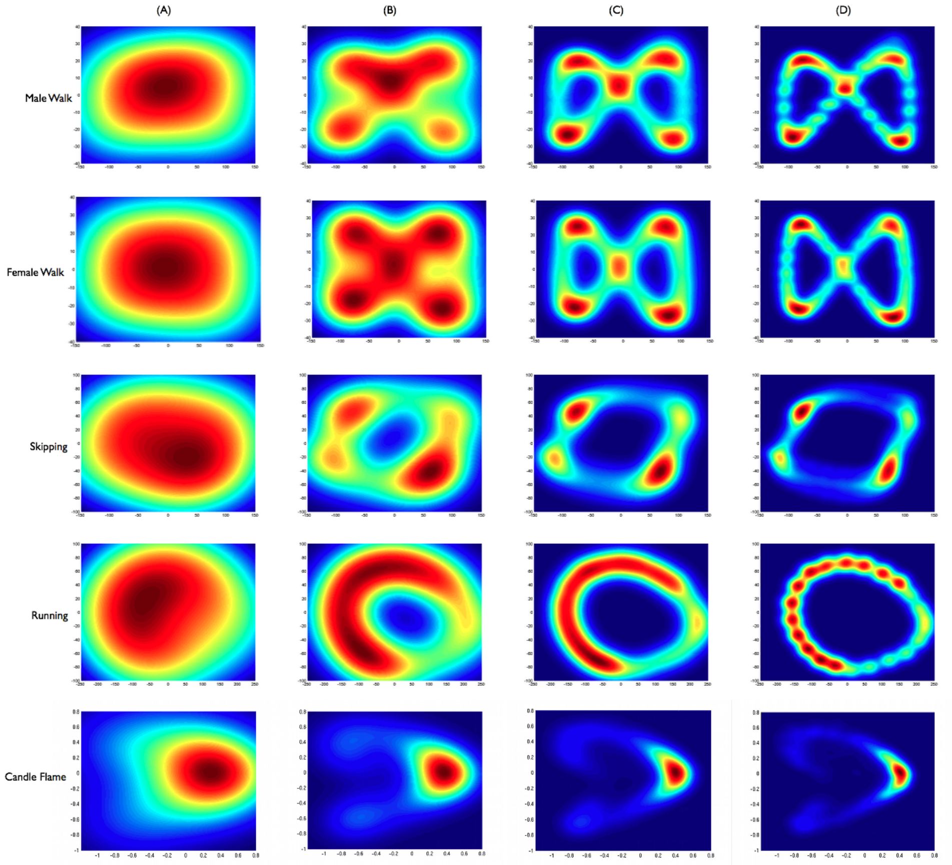


Figure 3.3: **Varying PDF kernel sizes:** Probability Density Function (PDF) for all sequences using different values of α. (A) $\alpha = 1$ (B) $\alpha = 0.25$ (C) $\alpha = 0.0625$ (D) $\alpha = 0.025$.

3.3.2 Fast Gaussian Approximation

As can be seen from Equation 3.3, the computation required for the probability density estimation is high since it requires an exhaustive calculation from the entire set of data examples. This would be too slow for a real-time implementation. The more samples used, the slower the computation, however, the more accurate the density estimation. As a result, a fast approximation method based on kd-trees [82] is used to reduce the estimation time without sacrificing accuracy.

Instead of computing kernel estimations based on all data points, with the kd-tree, queries are localised to neighbouring kernels, assuming the kernel estimations outside a local region contribute nominally to the density estimation. We are now able to specify N_n nearest neighbours to represent the model, where $N_n < N_T$. This significantly reduces the amount of computation required.

Equation 3.3 is simplified to:

$$P'(\mathbf{y}) = \frac{1}{|\mathbf{Y}'|} \sum_{\forall \mathbf{y}_i \in \mathbf{Y}'} p(\mathbf{y}_i) \tag{3.5}$$

where $\mathbf{Y}' \subseteq \mathbf{Y}$, and $\mathbf{Y}'$ is a set containing the N_n nearest neighbouring kernels to $\mathbf{y}$ found efficiently with the kd-tree.

3.3.3 Motion Synthesis

To generate novel motion the procedure is:

1. $P'(\mathbf{y})$ is constructed as PDF in the pose space that gives the likelihood of any particular pose configuration.

2. As we are particularly interested in motion, a second PDF is constructed that encodes the likelihood of motion in the pose space for a given configuration $P'(\mathbf{y}, \frac{d\mathbf{y}}{dt})$ where:

$$\frac{d\mathbf{y}_i}{dt} = \mathbf{y}_{i+1} - \mathbf{y}_i \tag{3.6}$$

   assuming regular sampling over the motion capture data. $P'(\mathbf{y}, \frac{d\mathbf{y}}{dt})$ is constructed similarly to Equation 3.5 using N_T Gaussian kernels in $\Re^{2d}$. Similarly to Equation 3.5, the covariance is set to:

$$\alpha \begin{pmatrix} \sqrt{\lambda_1} & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \sqrt{\lambda_d} & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \sigma_1 & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \sigma_d \end{pmatrix} \tag{3.7}$$

where σ_i is the standard deviation of the derivatives.

3. To locate a suitable starting configuration, the kernel that generates the highest likelihood is found.

$$max = \arg\max_{i=1}^{N_T}(P'(\mathbf{y}_i)) \tag{3.8}$$

4. From this configuration $\mathbf{y}_t = \mathbf{y}_{max}$, the highest likelihood movement is selected

$$max\Delta = \arg\max_{\forall \frac{d\mathbf{y}}{dt}} \left(P' \left(\mathbf{y}_t, \frac{d\mathbf{y}}{dt} \right) \right) \tag{3.9}$$

5. The model pose is then updated such that:

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \frac{d\mathbf{y}_{max\Delta}}{dt} \tag{3.10}$$

and the pose is then reconstructed for rendering as $\mathbf{x}_{t+1} = \mu + \mathbf{V}\mathbf{y}_{t+1}$.

6. The process then repeats from step 4.

3.3.4 Gradient Based Optimisation

This approach on its own will only generate the original motion. This is because currently, motion is guided only by the pre-computed derivatives with no optimisation to account for local gradient densities in each step. Therefore, optimisation is required to search for the local maxima at every iteration. To this end, *gradient based optimisation* is used to determine the most likely pose at every step.

Gradient based optimisation works well with the assumption that the optimisation in step 4 finds a good global maximum. Since the surface is smooth (due to the use of Gaussian kernels) and the space contiguous (in that the positions of two adjacent frames are spatially close in the eigenspace), a simple gradient ascent method can be used.

A Mean Shift approach works well since it is only necessary to asses the likelihood at the corners of the search region (again assuming the surface is smooth). However, such a search is $O(D^2q)$ where q is the number of iterations to convergence and D, the dimensionality of the space. It is worth noting that in the case of $P'(\mathbf{y}, \frac{d\mathbf{y}}{dt})$ this dimensionality is twice the number of eigenvectors retained in the projection. Line optimisation methods such as Powells Method work well as the surface is smooth and the search become linear in the number of dimensions $O(Dq)$. Newton type methods requiring Jacobian and Hessians are more problematic due to the large number of kernels in the PDF. Optimisation is therefore done along the direction of each eigenvector in turn, using the previous location as the starting point of the search.

3.3.5 Correction Term Constraint

With the addition of the gradient based optimisation, this model is capable of generating novel poses based on the PDF. Though gradient optimisation derives the local maxima, the pre-computed derivatives still dictate the global steps to the local regions. Since the derivatives are based on the original data with no knowledge of the kernel densities, there is an inherent risk that the derivatives can cause drifting out of range of the distribution. The gradient based optimisation can correct this error if the drift is minor. However, for more extreme cases, it may take several iterations and traversal through unrealistic poses before recovery.

To account for this drift, a *correction term* is added to the optimisation process. The *correction term* constrains the estimated posture to remain within the PDF. This is brought about by adding a weighting to all estimated postures. Each estimation is multiplied by the likelihood of that posture being a valid posture, discouraging movement outside the pose PDF. Step 4 therefore becomes:

$$max\Delta = \underset{\forall \frac{d\mathbf{y}}{dt}}{\arg\max} \left(P'\left(\mathbf{y}_t, \frac{d\mathbf{y}}{dt} \right) P'\left(\mathbf{y}_t + \frac{d\mathbf{y}}{dt} \right) \right) \tag{3.11}$$

This improves the optimisation in step 4, reducing drift, resulting in more plausible poses.

3.3.6 Naturalistic Variation using Gaussian Noise

Cyclic motion is characteristically periodic, and in most cases, repeats sinusoidally at an almost constant frequency. However, naturalistic cyclic motion, especially human movements, contain slight variations in different cycles. Ignoring these variances will result in identical motion cycles and produce unnatural looking animations. Currently, the model serves to generalise the motion but with no guarantee of the occurrence of naturalistic variation. In order to address this issue and encourage mild disparities between different cycles, *gaussian noise* is added to the pose estimation process.

A *noise term* $\mathbf{\Gamma}$ is calculated such that:

$$\mathbf{\Gamma}_j = \gamma \boldsymbol{\sigma}_j \psi, j \in \{1, ..., d\} \tag{3.12}$$

where γ is a normally distributed random number with $mean = 0$, $std = 1$, $\boldsymbol{\sigma}$ is the standard deviation of the derivatives, and ψ is *noise control parameter* that gives the user control of the noise magnitude. For this naturalistic variation to work effectively, the ability to vary $\mathbf{\Gamma}$ using ψ is crucial. If $\mathbf{\Gamma}$ is too large, it can inherently damage the animation by pushing the global traversals too far out of the normal cycle (and possibly completely out of the distribution), causing shaky and unnatural animations. Likewise, if $\mathbf{\Gamma}$ is too small, it will have little influence on the resulting animation. The *noise term* is added to the derivatives such that:

$$\frac{d'\mathbf{y}}{dt} = \frac{d\mathbf{y}}{dt} + \mathbf{\Gamma} \tag{3.13}$$

Step 4 now becomes:

$$max\Delta = \underset{\forall \frac{d'\mathbf{y}}{dt}}{\arg\max} \left(P'\left(\mathbf{y}_t, \frac{d'\mathbf{y}}{dt} \right) P'\left(\mathbf{y}_t + \frac{d'\mathbf{y}}{dt} \right) \right) \tag{3.14}$$

where $\mathbf{\Gamma}$ is re-computed at every iteration.

The addition of this *noise term* provides additional novelty to the generated animations, and is particularly beneficial when using smaller data sets that do not contain motion variation. The *noise term* is also useful when dealing with data sets with high variable densities. Such a data set can suffer from immobility if the global traversal cannot move far enough to optimise to a different local maxima, resulting in repetitive frames. In this case, the *noise term* helps to dislodge the global traversal to better local maximas.

The outcomes of varying $\mathbf{\Gamma}$ and the *noise control parameter* used for each data are presented in the result section.

3.4 Motion Blending using PDFs

Thus far, real-time motion synthesis is as a result of combining a kernel density estimation with a fast approximation method. To generate novel motion that follows an optimal path, a gradient based optimisation is used which incorporates correction term constraints to avoid drift. To improve the realism of the resulting animation, naturalistic motion variation is included to the motion derivatives, resulting in accurate, novel and realistic animations.

The approach to motion blending between different sequences using the PDF follows a very similar procedure to motion synthesis. The sequences are firstly projected into a combined eigenspace by performing PCA on their combined data (as explained in Section 3.2). A PDF is then constructed for each sequence. The PDF used in synthesis is the weighted average of all distributions. By changing the weighting W, the influence of each PDF on the final animation can be altered and transitions between the sequences made.

For instance, given $P'_a(\mathbf{y}_t)$ for pose PDF a, as presented in Equation 3.5, and $P'_b(\mathbf{y}_t)$ for PDF b, then step 3 is replaced with:

$$max = \arg\max_{i=1}^{N_T}([1-W]P'_a(\mathbf{y}_t) + WP'_b(\mathbf{y}_t)) \tag{3.15}$$

and similarly, step 4 is replaced with:

$$max\Delta = \arg\max_{\forall \frac{d'\mathbf{y}}{dt}} \left([1-W] P'_a\left(\mathbf{y}_t, \frac{d'\mathbf{y}}{dt}\right) P'\left(\mathbf{y}_t + \frac{d'\mathbf{y}}{dt}\right) + W P'_b\left(\mathbf{y}_t, \frac{d'\mathbf{y}}{dt}\right) P'\left(\mathbf{y}_t + \frac{d'\mathbf{y}}{dt}\right)\right) \tag{3.16}$$

where W is the weighting variable between 0 and 1 ($0 < W < 1$). $W = 0$ for PDF a resulting in the animation a, $W = 1$ for PDF b and $W = 0.5$ for the mid-point between them resulting in an animation of equal contribution from the PDFs.

3.5 Animation/Results

3.5.1 MoCap Walk Synthesis

To illustrate the approach, four MoCap sequences were used: *male walk*, *female walk*, *skipping*, and *running*. The sequences were captured from the same actor using 44 markers to cover the main joints of the human body (examples are shown in Figure 3.1 (A)). The markers are in 3D cartesian coordinates producing a 132 dimensional data sets (44(markers)×3(coordinates)). The sequences were projected down into their combined lower dimensional eigenspace of 5 dimensions using the approach detailed in Section 3.2. A model was constructed for each type of walk, where each model consisted of two PDF's, one for pose and one for motion. Combining the synthesis process in Section 3.3.3 with the gradient optimisation in Section 3.3.4, and by incorporating a correction term as discussed in Section 3.3.5, novel motion synthesis was possible.

Figure 3.4 column (A) shows a visualisation of the PDFs of four synthesised sequences projected onto the two primary eigenvectors. We will refer to the traversal of data points through their distribution as *motion trajectory*. These results are without the addition of a *noise term*. Using the outlined procedure, a maximum likelihood path was generated from each of the PDFs. The result of which are superimposed on top of their respective distributions. As can be seen, the data points corresponding to a sequence of postures remain within the distributions and follow the characteristic shapes of their original data. When animated, as shown in Figure 3.5, 3.6, 3.7, and 3.8, they resemble their original data and produce plausible animations.

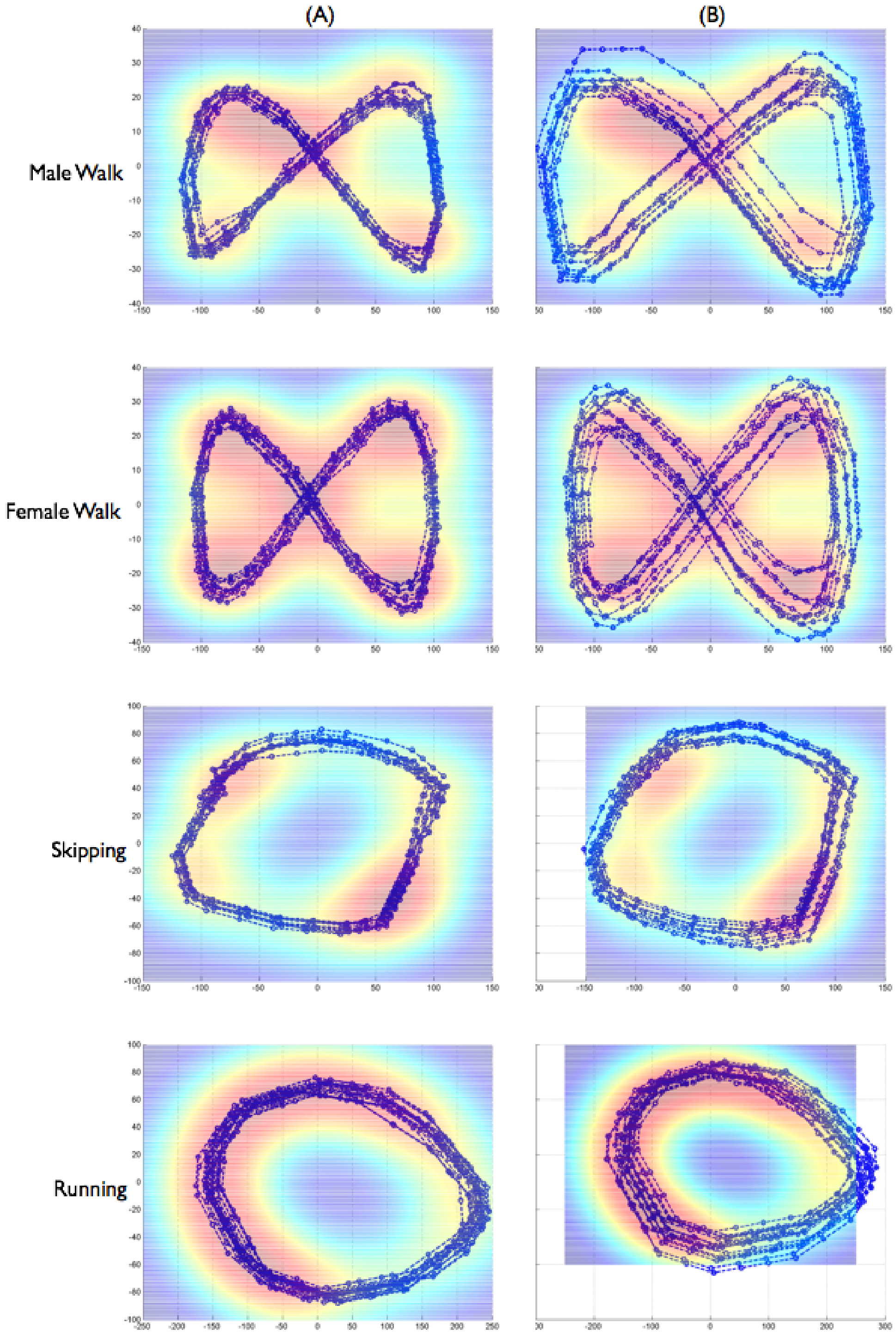


Figure 3.4: **Synthesis projections:** Plot of four synthesised motion capture sequences superimposed over their respective distributions. Column (A) is the synthesised results and column (B) is the synthesised results without using a correction term constraint.

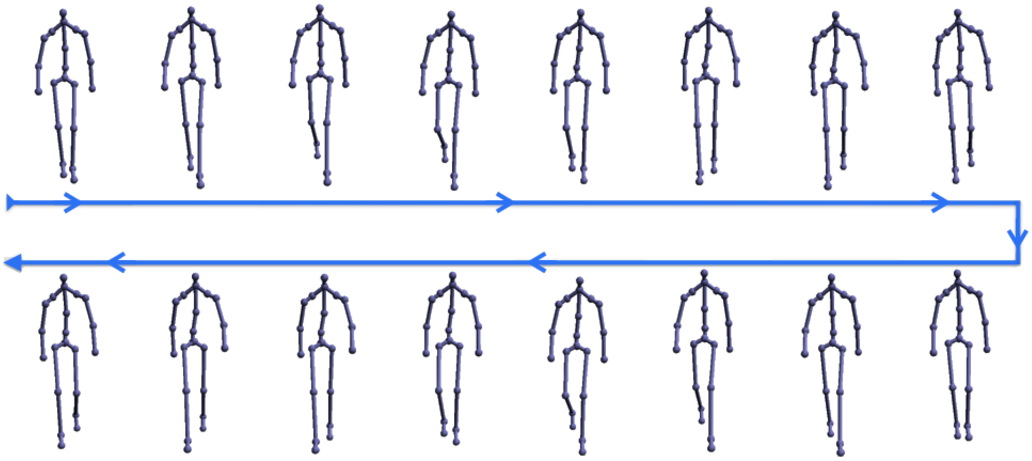


Figure 3.5: **Male walk:** Male walk synthesis as generated in Figure 3.4 column (A)

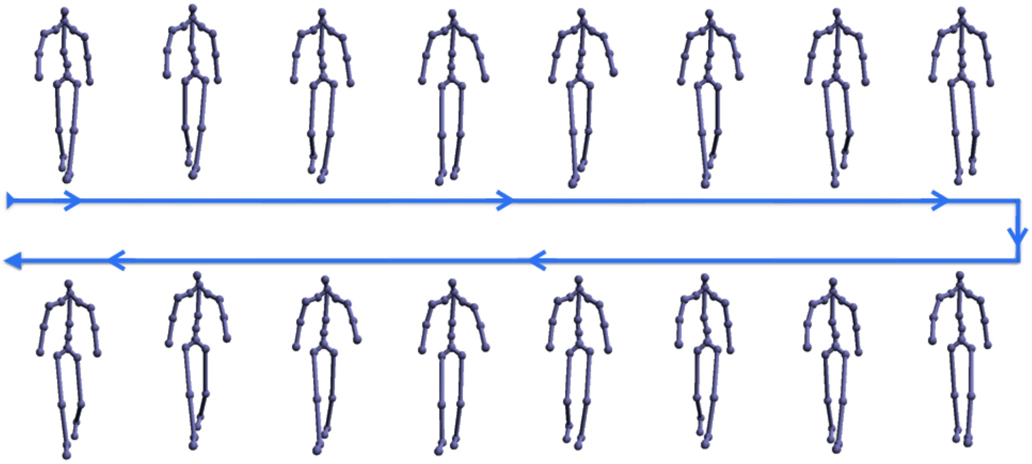


Figure 3.6: **Female walk:** Female walk synthesis as generated in Figure 3.4 column (A)

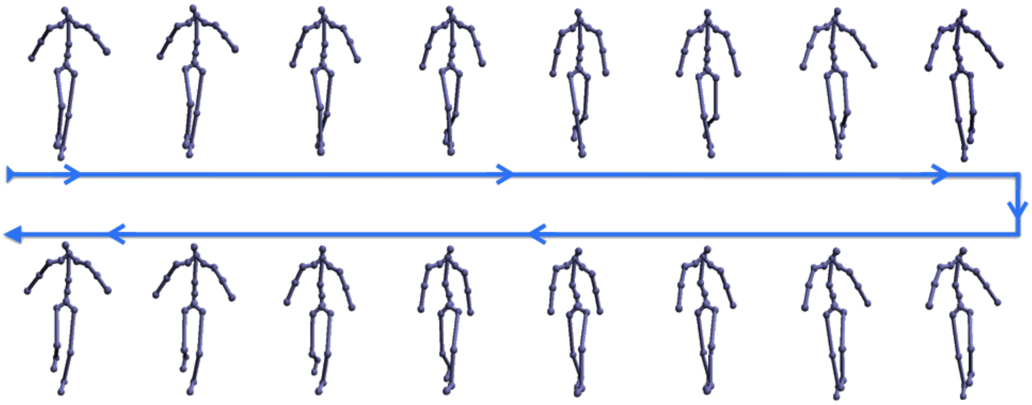


Figure 3.7: **Skipping:** Skipping synthesis as generated in Figure 3.4 column (A)

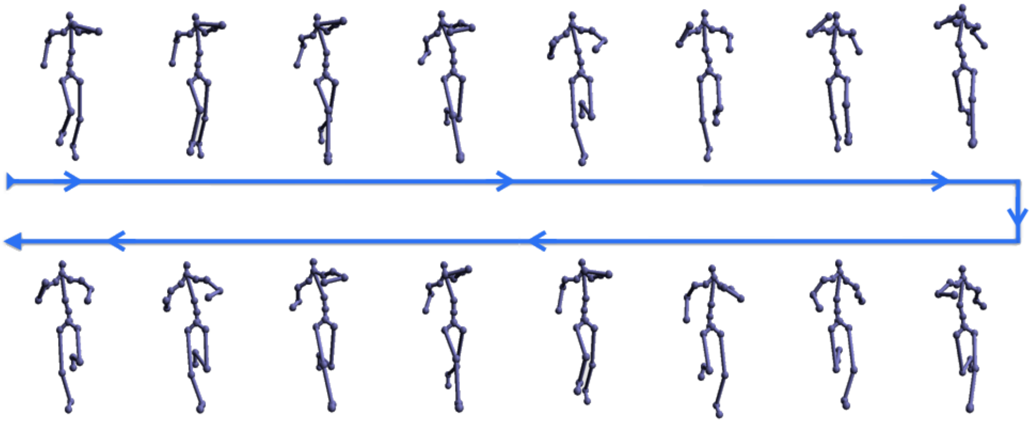
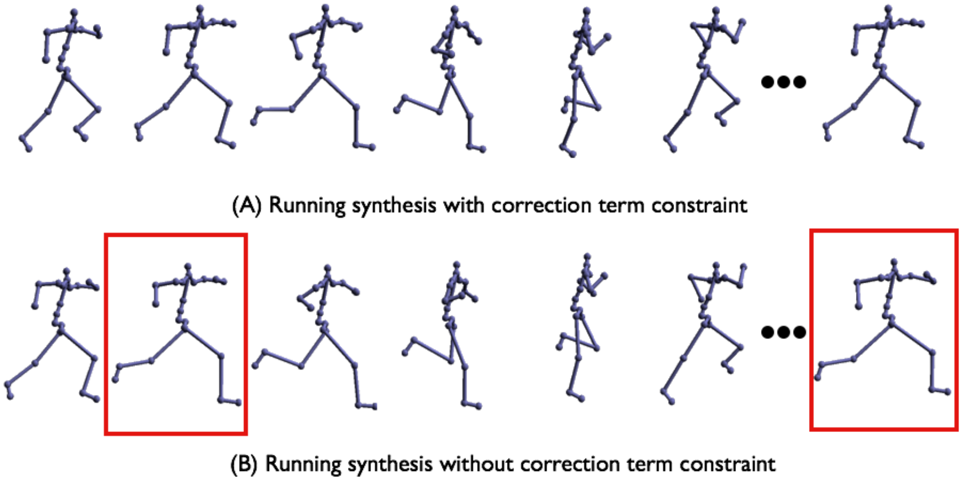


Figure 3.8: **Running:** Running synthesis as generated in Figure 3.4 column (A)



(A) Running synthesis with correction term constraint

(B) Running synthesis without correction term constraint

Figure 3.9: **Running based on correction term:** Running animation with (A) and without (B) correction term constraints. As highlighted by the red boxes, without the addition of the correction term constraint, the legs are unnaturally elongated. This occurs at points where the motion trajectory drops out of the distribution.

Figure 3.4 column (B), shows the outcome of not incorporating a *correction term constraint* to the synthesis process, as is the case in column (A). Again, this is without the addition of a *noise term*. The generated points in column (B) are sparse in comparison to column (A), not following the optimal path. These noisy motion trajectories are as a result of the optimisation error. At the outset, one can use the error as motion variation since it does produce mild variances in each cycle. However, the error is too unpredictable, and the influence of the noise cannot be controlled. Furthermore, in some cases, the optimisation error can result in the motion trajectory dropping out of the distribution, causing highly unnatural looking animations. This is most evident in the *running* and *skipping* results in Figure 3.4 column (B). Figure 3.9 shows the *running* animations with and without the correction term constraint. As highlighted by the red boxes in Figure 3.9 (B), without the addition of the correction term, the legs

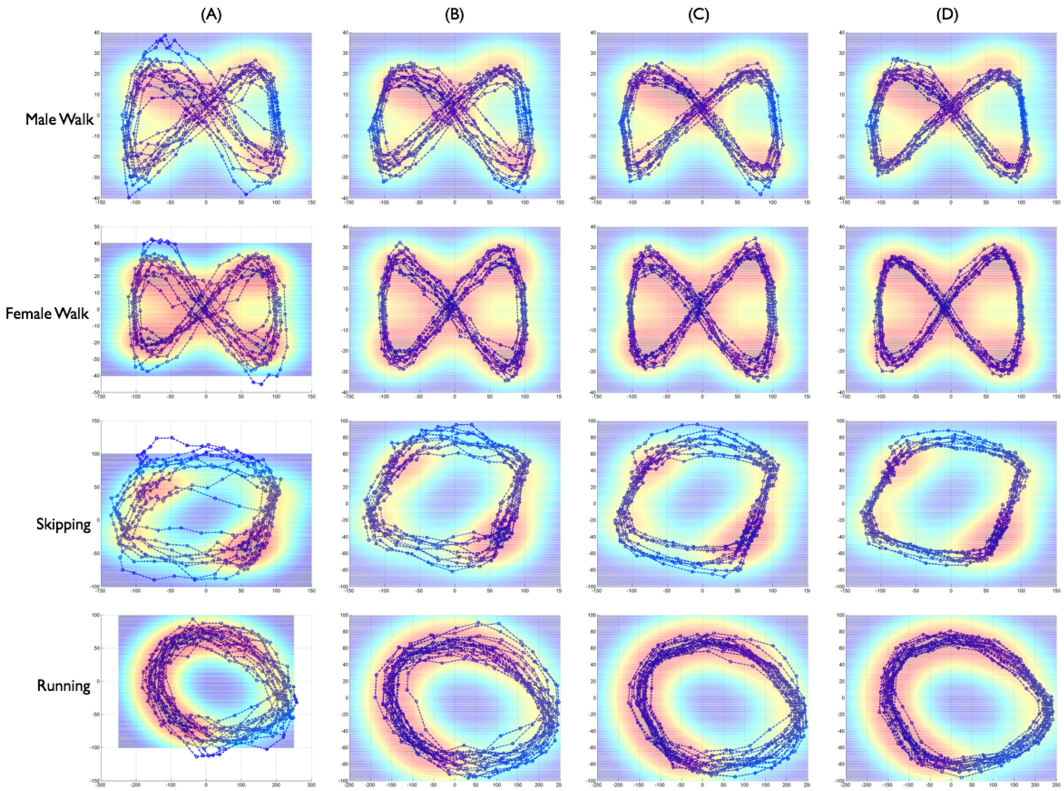
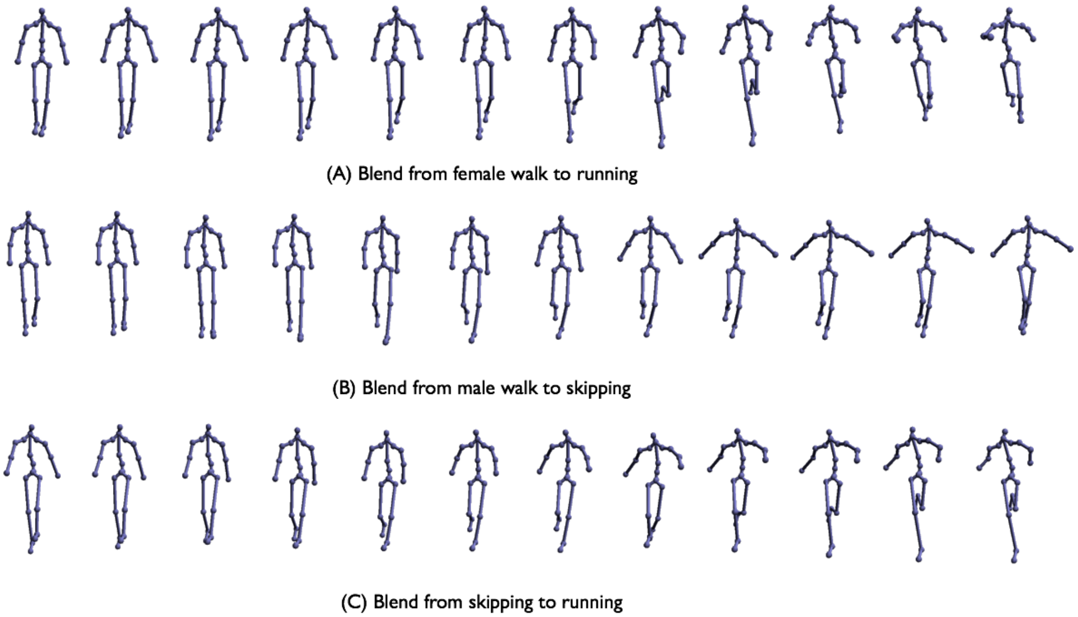


Figure 3.10: **Varying noise term:** Outcome of varying the *noise term* on synthesised motion trajectory using the *noise control parameter* ψ. (A) $\psi = 0.167$, (B) $\psi = 0.1$, (C) $\psi = 0.07$, (D) $\psi = 0.05$.



(A) Blend from female walk to running

(B) Blend from male walk to skipping

(C) Blend from skipping to running

Figure 3.11: **Blending walks:** Image showing animations of blended walks using PDF. (A) Blend from a female walk to running, (B) Blend from a male walk to skipping, (C) Blend from skipping to running

are unnaturally elongated. This occurs at points where the motion trajectory drops out of the distribution. As suggested earlier, the more reliable alternative is to reduce the optimisation error using the correction term constraint and apply the user controllable naturalistic variation to the synthesis process (as discussed in Section 3.3.6).

Figure 3.10 shows the outcome on the synthesised motion trajectory by varying the *noise control parameter* ψ. As shown in Figure 3.10 (A), if ψ is set too high, this results in jerky and unnatural animations. If ψ is set too low, as in Figure 3.10 (D), the resulting animations closely follows the optimal path with limited variability. For these MoCap experiments, $\psi = 0.07$ (as in Figure 3.10(C)) was used, which produced the ideal balance of realism and accuracy.

Figure 3.11 shows the effect of blending between motions by changing the weighting attributed to the contribution of any one PDF to the overall density estimate. Again it can be seen that smooth and natural transitions are achieved, even when moving between less usual motions such as *male walk* to *skipping*. These results were possible without the need for time-warping to establish frame correspondence, and whilst main-

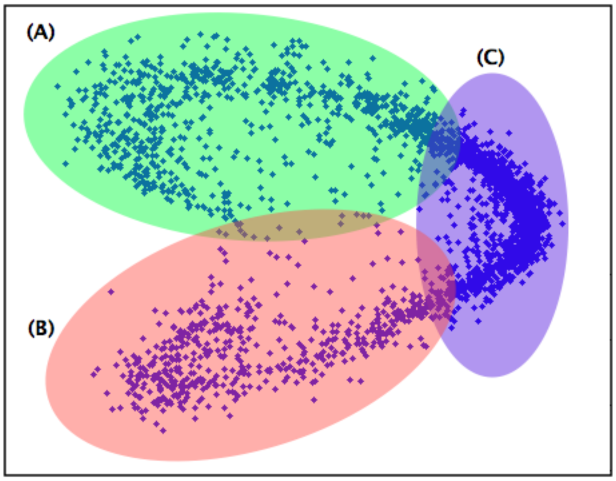


Figure 3.12: **Flame motion segmentation:** Automatic candle flame motion segmentation. (A) Flame blowing left region. (B) Flame blowing right region. (C) Stationary flame region.

taining the fundamental motion characteristics before, during and after transitions.

3.5.2 Candle Flame Video Texture Synthesis and Blending

The candle flame sequence was recorded using a webcam (185 × 140 pixels, 15 frames per second), lasting 3:20 minutes and containing 3000 frames. The recording was of a candle flame performing 3 different motions, *blowing left*, *blowing right* and burning in a *stationary position* (example of the data set is shown in Figure 3.1 (B)). The dimensional reduction process projected the data down to 42 dimensions.

As shown in Figure 3.12, the PCA projection of the first two principal components was used to automatically partition the video motion sequence into the three different motion types. Ellipse (A) correspond to frames of the candle flame blowing left, ellipse (B) blowing right, and ellipse (C) are frames corresponding to a stationary flame. A moderate overlap between the different partitions is permitted to assist with blending.

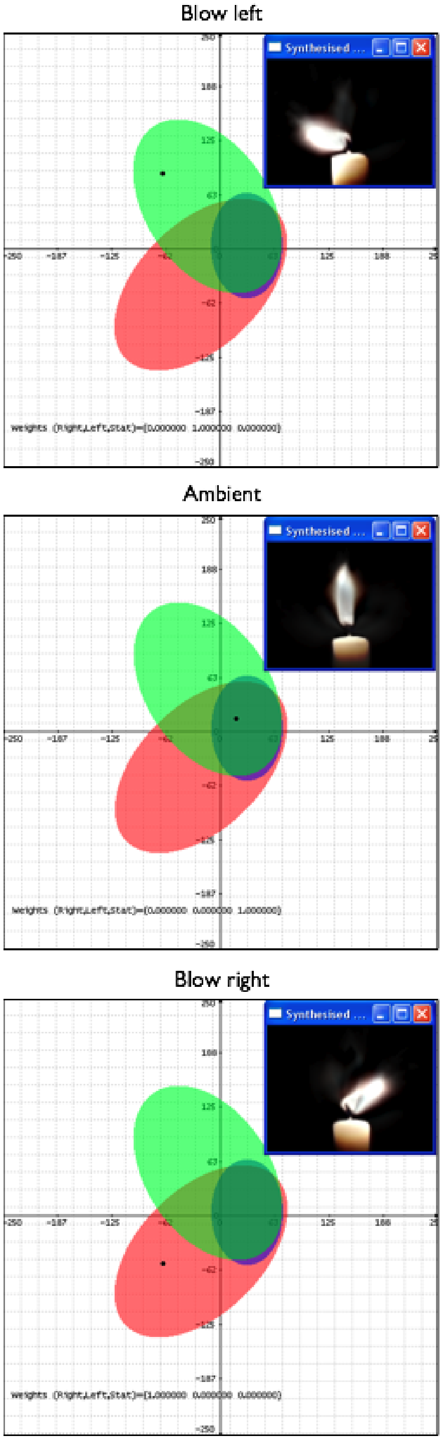
There is no limitations to the number of partitions that can be used, however, the best results are obtain when each partition relates to a specific motion type. If one attempts to overly partition within a motion type (e.g. flame blow state further partitioned to various angles of the flame blowing left), as well as limiting the potential for more varied movement within a motion type, when blending between PDFs, this increases the risk of transitioning through less densely populated or invalid regions in eigenspace resulting in blurred or inaccurate animations.

Akin to the MoCap data set, a model was constructed for each motion type, where each model consisted of two PDF's, one for pose and one for motion. Using the weighting variable as discussed in Section 3.4, the user can animate novel motion for each discrete candle state. The user is also able to blending between the different states in real time, maintaining the same motion characteristics as the original data. Results are shown in Figure 3.13. This shows the screen shots of the application in operation. It comprises of the synthesised video output and a graph. The graph contains three ellipses to provide a visual approximation of the positions of the different motion partitions. The black dot on the graph represents the current state in the synthesis process which is generated on the video output. As such, the graph enables the observation of the relationship between the motion partitions and the current state of synthesis.

It can be seen that the user has control over the discrete candle states which correspond accurately to the eigenspace partition of the first two principal components. Smooth blending is also plausible with natural variances, however with the same characteristics as in the original data set.

These results were obtained by including the *noise term* with $\psi = 0.5$. With regards to this data set, the addition of the *noise term* was essential for motion synthesis. As can be seen in Figure 3.2, the candle flame data set has a large variable density and a condensed distribution. This is most apparent when observing the distribution of discrete candle flame states as highlighted in Figure 3.12. The ambient flame state as shown in Figure 3.12 (C) is dense, representing very subtle changes in movement, which is characteristic of ambient motion. As such, the global motion estimation guided by the derivatives alone, is insufficient in traversing to a local gradient with a different local maxima. This can lead to a reduced mobility in generated motion variation, causing high frame repetition in the resulting animation. In this case, the *noise term* serves an additional purpose of dislodging global estimations to better optimise to other maximas.

As explained earlier, if the *noise term* is not included, or set too low, the resulting



Blow left

```
Weights (Right,Left,Stat)=(0.000000 1.000000 0.000000)
```

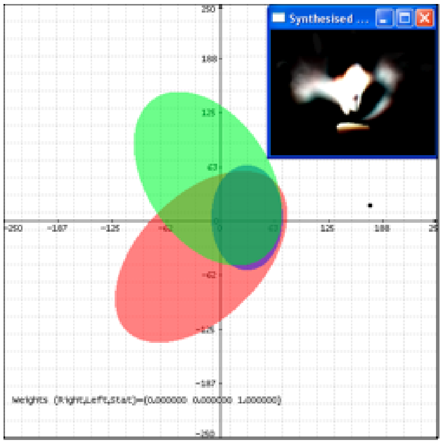
Ambient

```
Weights (Right,Left,Stat)=(0.000000 0.000000 1.000000)
```

Blow right

```
Weights (Right,Left,Stat)=(1.000000 0.000000 0.000000)
```

Figure 3.13: **Flame synthesis:** Screen shots of candle flame synthesis application in operation. The generated output for synthesising *flame blow left*, *flame blow right*, and *ambient flame* are shown separately.



Weights (Right,Left,Stat)=(0.000000 0.000000 1.000000)

Figure 3.14: **Flame synthesis with noise:** Candle flame synthesis with very high *noise term*. In this example $\psi = 3$.

animation may contain several repetitive frames and appear unnatural. Demonstrated in Figure 3.14, if the *noise term* is too high ($\psi = 3$), the generated motion can drop out of the distribution and skew the animation.

Shown in Figure 3.15, when comparing the synthesised output to the original data, there exists a slight *ghosting effect* of the flame in the synthesised video which is most apparent during blending. Several contributing factors have caused this. Firstly, the video data set was dimensionlly reduced using PCA. Though the removed principal components combine to make up only less than 5% of the total variance, their absence can cause blurring in the reprojected video sequence. Secondly, since the generated frames during synthesis are only a generalisation of the motion rather than the original data, the estimated frames themselves can be obscure. This is especially the case when blending between different PDFs, since blending involves estimating frames between different motion types, which in most cases, are not a prominent occurrences in the original data.

Such ambiguities are more evident in video synthesis than in MoCap synthesis. The estimated poses using the MoCap data set are cartesian coordinates. Hence, these inconsistencies can only affect the articulated movements which is less obvious than the resulting inconsistencies in pixel intensities.

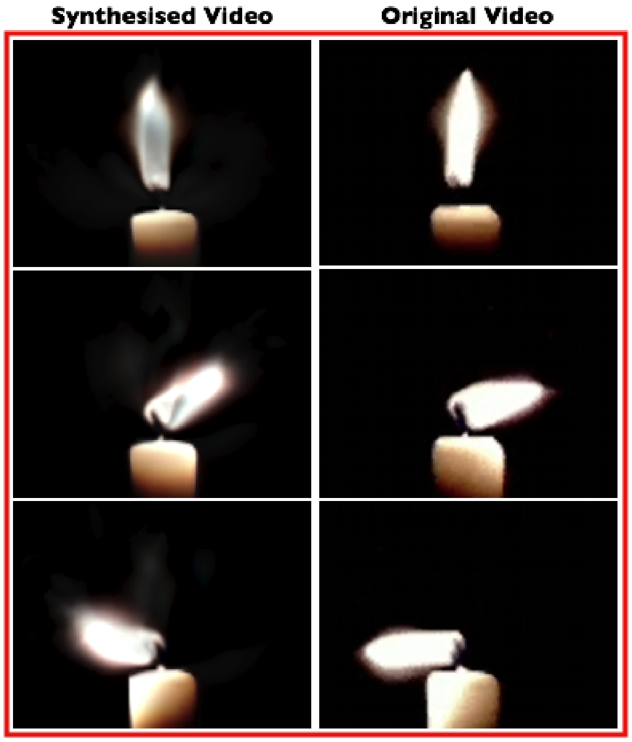


Figure 3.15: **Ghosting effect:** Example of *ghosting effect* in video data set

3.6 Summary

This chapter presents a generative model for motion synthesis and blending using PDFs. The model is applicable to both MoCap and video texture data sets, capable of synthesising novel motion with the same characteristics as the original data. By combining the model with a fast Gaussian approximation method, real-time density estimation and motion synthesis is possible. By also incorporating a gradient based optimisation with a correction term constraint, motion synthesis follows the optimal path as presented by their respective PDFs. With the addition of naturalistic variation, more realistic motion with mild dissimilarities in each cycle is achieved. As the results suggest, as long as there is a complete motion cycle, synthesis is possible regardless of the quantity of frames in the data set. Additionally, as no time alignment is required for the process, the temporal information in the animation is preserved.

However, this approach for synthesising and blending is best suited to cyclic MoCap and stochastic videos, and would result in less natural looking blends if attempted on more complex and specific motion. This is mainly due to the motion generalisation of the approach, whereby more complex motion may not generalise well. The *ghosting effect* in the texture synthesis process is also a limiting factor, since texture generalisation cannot guarantee aesthetically pleasing video animations.

Another limitation is present in the linear blending of PDFs for the purpose of motion transitions. This can only work if the different motion types for the respective data set, partially or completely overlapping each other in eigenspace. This is mostly the case for similar cyclic motion. If no overlap exists, linear blending of PDFs is as a result of travelling through an unknown region in eigenspace which will result in unnatural looking motion transitions. This is caused by the unknown region in eigenspace not adhering to any modelled motion constraints.

Example-based methods to motion synthesis would provide a more suitable alternative. By retaining the original motion to use in synthesis, there is no loss of detail from the original data. Additionally, by segmenting the motion data into short subsequences with *start* and *end* transition points, transitioning between more complex motion types with minimal or no overlap is made possible.

The next chapter presents a novel example-based approach to motion synthesis, addressing these limitations.

Chapter 4

MIMiC: Multimodal Interactive Motion Controller

The use of statistical models to generalise motion characteristics works well for synthesising simplistic and periodic motion. However, the drawback to this approach is there is no assurance of the quality of the animation produced, as information from the data set is lost through the blurring of the PDFs necessary for generalisation. Example-based methods to motion synthesis retains the original motion data to use in synthesis. Although the ability to produce novel generalised animations of the data is minimised, it provides a lossless alternative capable of handling more complex motion characteristics.

This chapter presents a novel example-based method to motion synthesis. Using an unsupervised motion segmentation approach, appropriate points for seamlessly transitioning between different subsequences of the original data are obtained. As in Chapter 3, the likelihood of a pose is modelled as a pose space Probability Density Function (PDF). Instead of precomputed derivatives, a Markov Transition Matrix is used to derive the probability of motion and apply additional constraints to the motion dynamics. As such, novel motion synthesis is as a result of computing the probability of transitioning from one subsequence to another. Additionally, by learning the mapping between motion subspaces and external stimulus, the user can drive the motion at an intuitive level, giving the user real-time interactive *Multimodal Control* of the creation

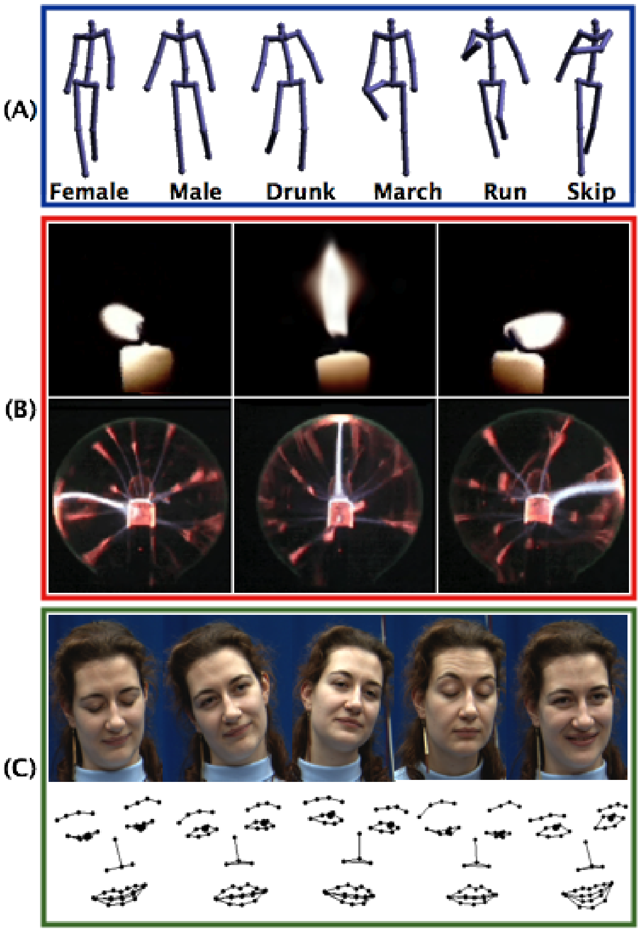


Figure 4.1: **Data set:** (A) Sample motion captured data of different types of walks, (B) Candle and plasma beam recorded whilst undergoing motion, (C) Tracked face data used in modelling conversational cues.

of novel sequences. The external stimulus could come from any modality and the use of auditory, touch and gesture is demonstrated within this work. Combining the real-time *Motion Model* with interactive *Multimodal Control*, we get the *Multimodal Interactive Motion Controller* (MIMiC), giving a user multimodal control of motion data of various formats.

Figure 4.1 shows the example applications. Figure 4.1 (A) shows six different types of motion captured walks. Using MIMiC, a user is able to generate novel movement and transitions between different types of cyclic motion such as running and skipping, and to even more complex motion types such as drunken walking. This MoCap data set is different from that used in Chapter 3. These new sequences were chosen specifically because they contain more frames with greater variability. Since this example-based approach uses the actual data for synthesis and not a statistical approximation (as performed in Chapter 3), a *noise term* (explained in Section 3.3.6) cannot be used to stimulate variation. All generated animations are embedded within the data hence, MoCap data with more variation is needed.

Figure 4.1 (B) shows example frames from video sequences of a candle flame and plasma ball used as video textures. The candle flame sequence is identical to that used in Chapter 3. Here, the purpose of MIMiC is to control the direction in which the flame and beam move in real-time, whilst generating a novel animation with plausible transitions between different types of movement. Since the animation is a replay of the original data and not a generalisation, there is no *ghosting effect* in the video animations (as previously highlighted in Figure 3.15), resulting in clearer and more accurate rendering.

MIMiC also permits the modelling of highly elaborate facial movements like *nodding* and *blinking*, which takes place during naturalistic conversations. As shown in Figure 4.1 (C), this is demonstrated on a 2D tracked contour of a face generated from a video sequence of a person listening to a speaker. Mapping the audio features of the speaker to the 2D face, the model generates appropriate non-verbal responses triggered from audio input.

The chapter is divided into the following sections. Section 4.1 details an overview of the MIMiC system. Sections 4.2 presents the data set used for motion modelling and their respective multimodal control interfaces. Sections 4.3 describe the process of building the dynamic model. Section 4.4 presents the interactive multimodal controller, and the remainder of this chapter describes the results and summary.

4.1 Overview

MIMiC allows a user to reproduce motion in a novel way by specifying, in real-time, which type of motion inherent in the original sequence to perform. As shown in Figure 4.2, the system comprises two stages: learning a *Motion Model*, and building a

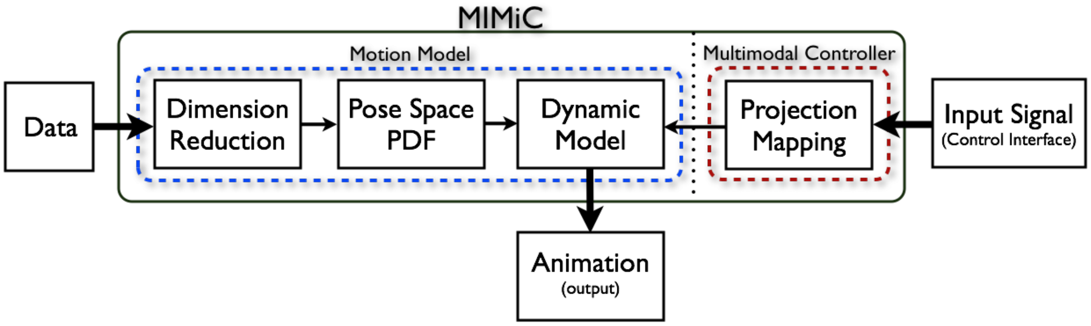


Figure 4.2: **System Overview:** Flow chart of MIMiC system. Consists of two main stages, the *Motion Model* and the *Multimodal Controller*. The *Motion Model* takes a data set and creates a dynamic model of motion. The *Multimodal Controller* uses projection mapping to translate user commands from an input signal to the dynamic model. The system generates the desired output as synthesised novel animations

Multimodal Controller.

The process of learning a *Motion Model* starts very similarly to the generative model in Chapter 3. It begins with the data which is the input to the system. The data can be of various formats (see Section 4.2). Given the data, eigenspace decomposition is used to reduce the dimensionality to a lower dimensional space as explained in Section 3.2. We will refer to this lower dimensional space as *pose space*. Figure 4.3 row (A) shows plots of the different data sets in *pose space* projected onto the first two eigenvectors.

Conventional example-based methods for motion synthesis such as *motion graphs* [101, 111, 13, 17, 70], traverses a graph, connecting motion segments based on user specified constraints such as position, orientation and timing. Little interest is given to how common or likely the connecting nodes are given the data set. However, better quality transitions can be produced by computing the likelihood of a pose or frame as an additional parameterised weight. Hence, a statistical model is learnt to derive the likelihood of pose in *pose space* based on the respective data set. Using the kernel density estimation in Section 3.3.1, combined with the fast approximation methods in Section 3.3.2, a *pose space PDF* is learnt.

As opposed to building a second PDF to derive the likelihood of motion (as described in the synthesis process in Section 3.3.3), a dynamic model is encoded. The encoding

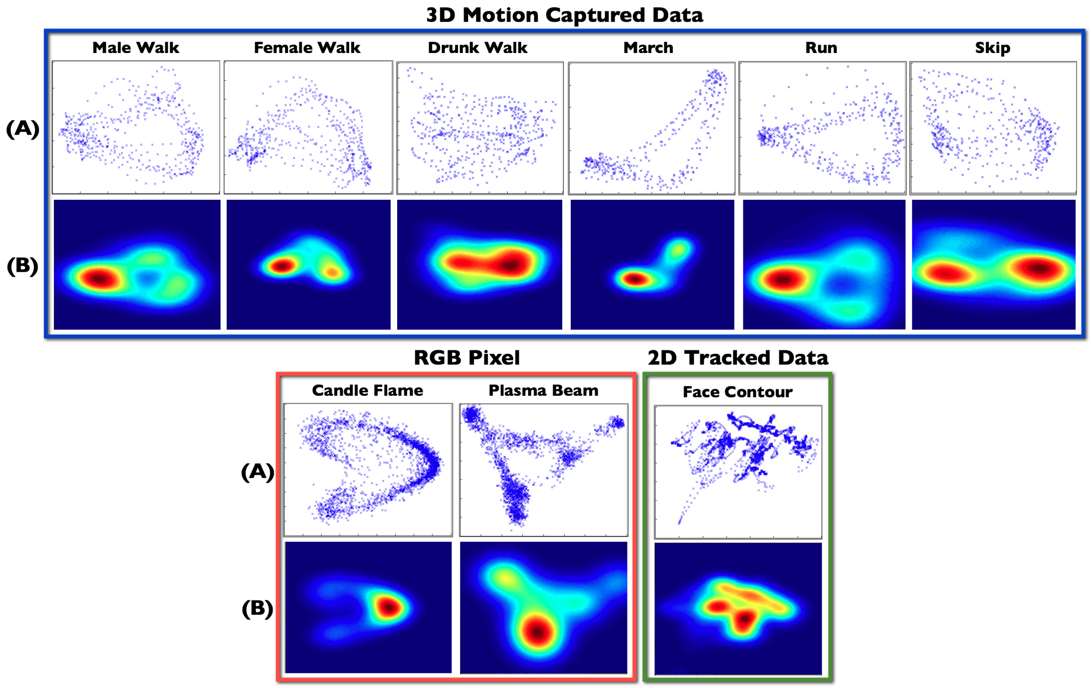


Figure 4.3: **PCA projection and PDFs:** Row (A): Plot of eigen projections of the first 2 principal components of all data sets. Row (B): PDF of pose space where the kernel size has been scaled by $\alpha = 0.25$.

consists of an unsupervised segmentation method to derive cut point clusters, whereby each cluster represents groups of similar frames that can be seamlessly blended together. These cut points are used as transition points, through which the set of consecutive frames between adjoined transition points make up subsequences. A first-order Markov Transition Matrix is learnt by treating each cut point cluster as a state in a Markov process. Motion is generated as high likelihood transitions from one subsequence to another based on the *pose space PDF* and the probability of the given transition determined by the Markov Transition Matrix.

The second stage is the *Multimodal Controller* which allows real-time manipulation of the *Motion Model* based upon an input signal. The controller consists of a *projection mapping* between the model and input signal, which reweights the *pose space PDF* to produce the desired movement.

4.2 Data Set for MIMiC

Given a motion sequence $\mathbf{X}$, each frame is represented as a vector $\mathbf{x}_i$ where $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{N_T}\}$ and N_T is the number of frames.

Various motions can be modelled by the system. We demonstrate 3D motion, 2D tracked points, and rgb pixel intensities in four examples:

- **3D MoCap Data:** The user can specify in real-time which type of animated walk to generate. By requesting a set of different walks, the system can blend between them whilst retaining the natural variance inherent in the original data. Six different walks are used: *male walk*, *female walk*, *drunk walk*, *march*, *run*, and *skip*.
- **Candle Flame:** The movement of a candle flame is synthesised where the user has control over three discrete states: *ambient flame*, *flame blow left*, *flame blow right*, and can blend between them. Using simple computer vision, the user can perform hand-waving gestures to influence the direction of the flame, giving the illusion of creating a draft/breeze that influences the animation.
- **Plasma Beam:** The user controls the movement of a plasma beam using a mouse cursor or a touch screen monitor. The plasma beam responds to the user's touch in real-time.
- **Tracked 2D Face Contour:** An animation of a 2D face is driven directly from audio speech signals, displaying appropriate non-verbal visual responses for an avid listener based on a speaker's audio signal.

In all cases, each time step i of the data to be modelled is vectorised as $\mathbf{x}_i = (x_{i1}, y_{i1}, ..., x_{ia}, y_{ia}) \in \Re^{2a}$ for a 2D contour of a points, $\mathbf{x}_i = (x_{i1}, y_{i1}, z_{i1}, ..., x_{ib}, y_{ib}, z_{ib}) \in \Re^{3b}$ for a 3D contour of b points and $\mathbf{x}_i = (r_{11}, g_{11}, b_{11}, ..., r_{\mathrm{xy}}, g_{\mathrm{xy}}, b_{\mathrm{xy}}) \in \Re^{3\mathrm{xy}}$ for an x × y image.

As explained in Chapter 3, Section 3.2, PCA is used for dimension reduction, resulting in a d-dimensional representation of each frame in the sequence. This representation

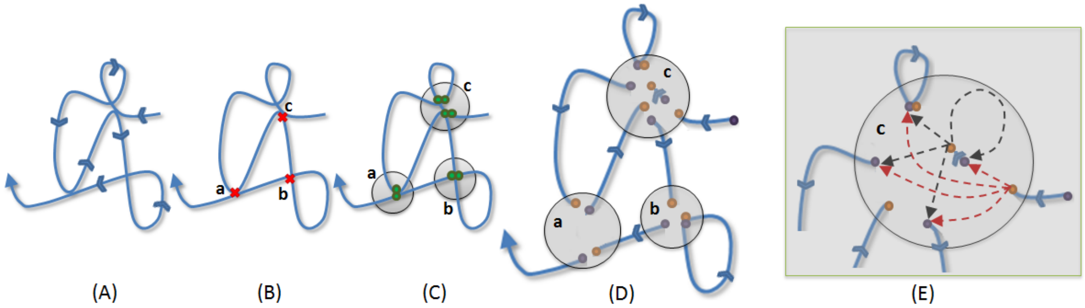


Figure 4.4: **Unsupervised motion segmentation:** (A) Trajectory of the original motion sequence. Arrows indicate the direction of motion. (B) $N_c = 3$ k-medoid points derived using the unsupervised k-medoid clustering algorithm. The three red crosses are the three k-medoid points a, b, and c. (C) The small green dots are the cut points derived as the nearest neighbouring points to a k-medoid point less than a user defined threshold θ. The three gray circles represent cut point clusters a, b and c. (D) Cut points act as start and end transition points segmenting the data into shorter segments. The orange dots are start transition points and the purple dots are end transition points. (E) Diagram of possible transitions within cluster c. For simplicity only a few transitions are displayed.

reduces the computational and storage complexity of the data whilst still retaining the time varying relationships between each frame. Figure 4.3 (A) shows a plot of the different data sets projected onto the first two eigenvectors.

4.3 Dynamic Model

By learning a PDF, the data is represented in a generalised form which is analogous to a generative model (see Chapter 3). From this it is possible to generate novel motion, using pre-computed motion derivatives, combined with a gradient decent for optimisation. However, such a model runs the risk of smoothing out subtle motion details, and is only suited for simple motion. To overcome these limitations, we segment the original data into shorter subsequences, and combine the PDF with a Markov Transition Matrix to determine the likelihood of transitioning to a subsequence given a pose configuration. This allows motion generation based on the original data, retaining subtle but important motion information. It also allows our motion model to work with non-periodic motion data.

The reminder of this section is divided into four parts. First, the unsupervised segmentation approach is described. In the following three sections, the Markov Transition Matrix, the generation of novel motion sequences, and the dynamic programming method for forward planning are explained respectively.

4.3.1 Unsupervised Motion Segmentation

Similar to most work on motion synthesis, the motion data needs to be analysed to compute some measure of similarity between frames and derive points of intersection within the data. These points are used to segment the motion data into several short subsequences, where a single subsequence is represented as a set of consecutive frames between a start and end transition point. The idea is to connect various subsequences together to create a plausible novel sequence.

The common approach is to compute the L2 distance over a window of frames in time and use a user defined threshold to derive points of intersection within the data to use as transition points [70, 44, 106, 101, 111]. This approach works well, however, for large data sets, it can be tedious to compute the distance between every frame. Balci et al. [13] proposed an iterative clustering procedure based on k-means to define clusters of poses suitable for transitions. However, k-means produces cluster centres not embedded in the data which can result in noise and outliers. Instead, in this work a k-medoid cluster algorithm is adopted to define N_c k-medoid points, where $N_c < N_T$. Each k-medoid point is defined as the local median in regions of high density, and can be used to define regions where appropriate transitions are possible. By only computing the L2 distance at these points, the amount of computation required to define candidate transitions is reduced, focusing attention on regions where transitions are most likely.

Figure 4.4 shows an example of the process. Given a motion sample, shown by the two dimensional motion trajectory in eigenspace in Figure 4.4 (A), a k-medoid clustering algorithm is used to find N_c k-medoid points. We define each k-medoid point as $\boldsymbol{\delta}_n^c$ given by the k-medoid method whereby $\boldsymbol{\delta}_n^c \in \mathbf{Y}$. In Figure 4.4 (B), $N_c = 3$ and are shown as the three red crosses which we refer to as a, b, and c.

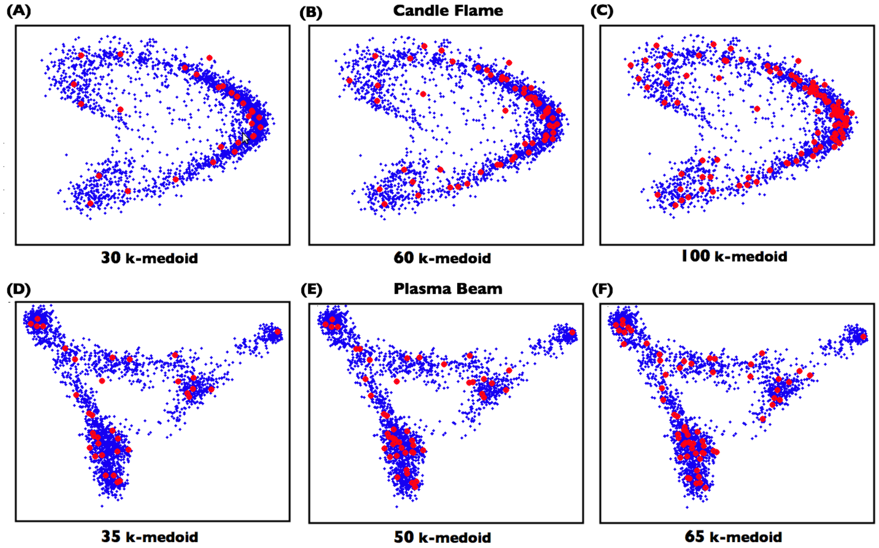


Figure 4.5: **K-medoid distributions:** Plot showing the distributions of varying numbers of k-medoids relative to the data set. (A), (B), and (C) relate to the candle flame data set, and (D), (E), and (F) to the plasma beam data set. The blue points are the eigen projections of the first 2 principal components, and the red points are the k-medoids.

N_c is empirically determined based on the number of clusters that best defines the distribution of poses in pose space. This is demonstrated in Figure 4.5, showing the distributions of varying numbers of k-medoids relative to the data set. Figure 4.5 (A) and (D) shows a low distribution of 30 and 35 k-medoid points for the candle flame and plasma beam data set respectively. Though the most densely populated areas have sufficient distribution of k-medoids points, the less densely populated areas do not. As a result, N_c is increased until a satisfactory distribution in pose space has been reached. Shown in Figure 4.5 (C) and (F), a high distribution of 100 and 65 k-mediods for the candle flame and plasma beam data set respectively, presenting a better spread of k-medoids across the respective data sets.

The outcome of varying this parameter is qualitative. N_c is not sensitive to small variations, having a large range over which it makes little difference to animation. However, if N_c is too high, the model will generate unrealistic motion as a result of shorter motion subsequences causing highly frequent and unnatural transitions. If N_c is too low, the subsequences will be too long, reducing the novelty of animation and the model's responsiveness to user commands. Different values of N_c were chosen for the different data sets based on this condition, and are detailed in Section 4.5.

Using a user defined threshold θ, the nearest points to each k-medoid points are identified to form clusters of cut points. The cut points are represented by the small green dots in Figure 4.4 (C), and the clusters of cut points are represented by the gray circles. The set containing the cut points of the n^{th} cluster is defined as $\mathbf{Y}^c_n = \{\mathbf{y}^c_{n,1}, ..., \mathbf{y}^c_{n,Q_n}\}$, where the number of cut points of the n^{th} cluster is denoted as Q_n.

Threshold θ provides the user with a tolerance on how close cut points need to be in eigenspace to form a valid transition. It is determined experimentally whereby if it is set too high, it becomes more challenging to produce plausible blends when making transitions, and if too low, potential cut points are ignored and we are limited to points that overlap, which is an unlikely occurance in a multi-dimensional space. This is demonstrated in Figure 4.6. The graphs show the L2 distance between a k-mediod point and all points in the data set. There are two examples for the candle flame data set (Figure 4.6 (A) and (B)), and the plasma beam data set (Figure 4.6 (C) and (D)). The red line across the graph represents the chosen threshold θ, and the red dots are the chosen cut points. These cut points are selected as local minima below θ. The value of θ for each data set (presented in Section 4.5) where chosen to set an acceptable trade-off between having good transitions (low threshold) and having high connectivity (high threshold).

The cut point clusters consist of discrete frames which are not directly linked, however smooth transitions can be made between them. Simple blending techniques such as linear interpolation can reliably generate a transition. The simplicity of linear interpolation also allows for quick computation, supporting real-time animation rendering during motion blending. As shown, in Figure 4.4 (D) the cut points are used to segment the data to smaller subsequences with start and end transition points. Figure 4.4 (E) shows a few of the possible transitions between various subsequences in cluster c.

In cases where the recovered cut point clusters in pose space are sparsely populated, they are automatically pruned and removed from the network of clusters.

Shown in Figure 4.7, for simplicity, the transitions from the n^{th} cluster contents are

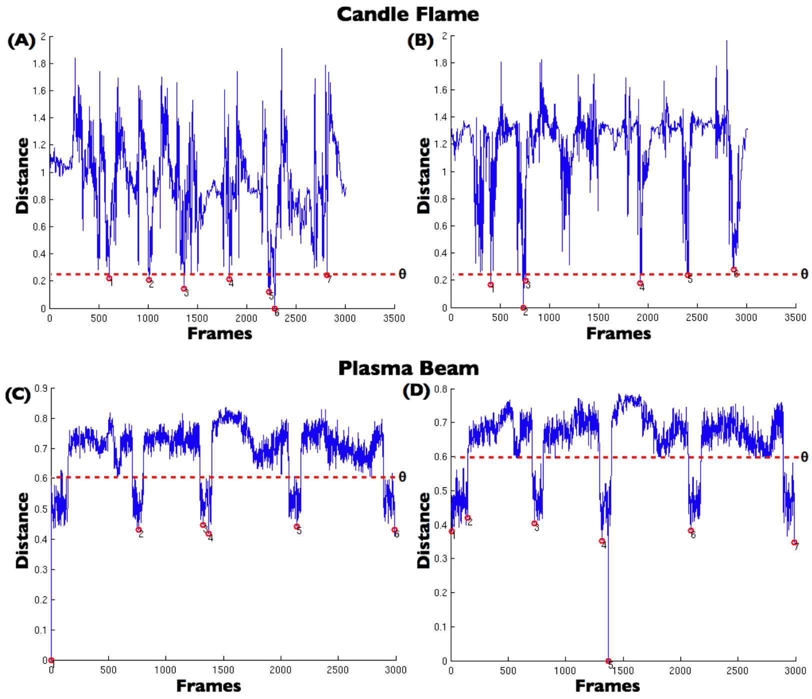


Figure 4.6: **Distances from k-medoid:** Plot showing two examples of the L2 distance between a k-medoid point and all points in the data set, for the candle flame data set ((A) and (B)), and the plasma beam data set ((C) and (D))

defined as the triplets $\{(\mathbf{y}^c_{n,1}, \mathbf{z}^c_{n,1}, C^z_{n,1}), ..., (\mathbf{y}^c_{n,Q_n}, \mathbf{z}^c_{n,Q_n}, C^z_{n,Q_n})\}$, where $\mathbf{y}^c_n$ is a cut point in the n^{th} cluster acting as the start transition point, $\mathbf{z}^c_n$ is the end transition point denoting the end of the subsequence between $\mathbf{y}^c_n$ and $\mathbf{z}^c_n$, where $\mathbf{z}^c_n \in \mathbf{Y}^c$ and $\mathbf{z}^c_n \neq \mathbf{y}^c_n$, and C^z_n is the index of the cluster $\mathbf{z}^c_n$ belongs to. In this example, $Q_n = 3$.

In most of our data sets, there are variable densities across different motion types. A specific example is the candle flame data set which has a heavy bias towards the stationary flame state due to the quantity of data acquired for each state. This is evident in Figure 4.3. The k-medoid algorithm attempts to find exemplars which cover the entire manifold/subspace of data points. Adding more of the uncommon motion types/animation to the data set is also possible as k-medoid will attempt to evenly partition the entire data set.

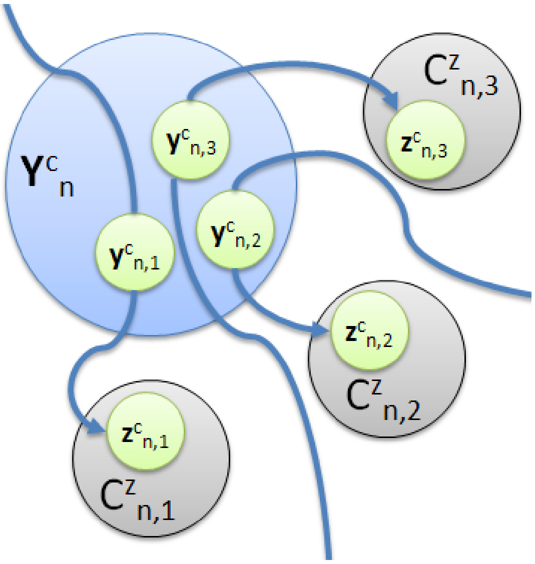


Figure 4.7: **Transitions between cut points:** Example of transitions between cut points in different clusters. The six green circles are cut points and the blue lines are sets of consecutive frames connecting them. These sets of consecutive frames make up the different subsequences. The blue circle $\mathbf{Y}^c_n$, represents the cluster of start transition points, and the three grey circles indexed as C^z_n, represents the clusters of end transition points

4.3.2 Markov Transition Matrix

When generating novel motion sequences, we are not only interested in generating the most likely pose but also the most probable path leading to it. A first order Markov Transition Matrix [54] is used to discourage movements that are not inherent in the training data. As an approach formally used with time-homogeneous Markov chains to define transition between states, by treating our clusters of cut points $\mathbf{Y}^c_i$ as states, this approach can be used to apply further constraints and increase the accuracy of the transition between sequences. An example of this is shown in Figure 4.8 for the case of the candle flame data set, depicting the k-medoid frames at each of the states, and the transition probabilities between them. Going from a blow left state straight to a blow right state is not a probable transition, instead going to a station flame state first then to a blow right state is more probable, and in most cases, a better looking motion transition.

We define $\mathbf{P} = \{p_{k,l}\}$ as the transition matrix whereby $p_{k,l}$ denotes the probability of

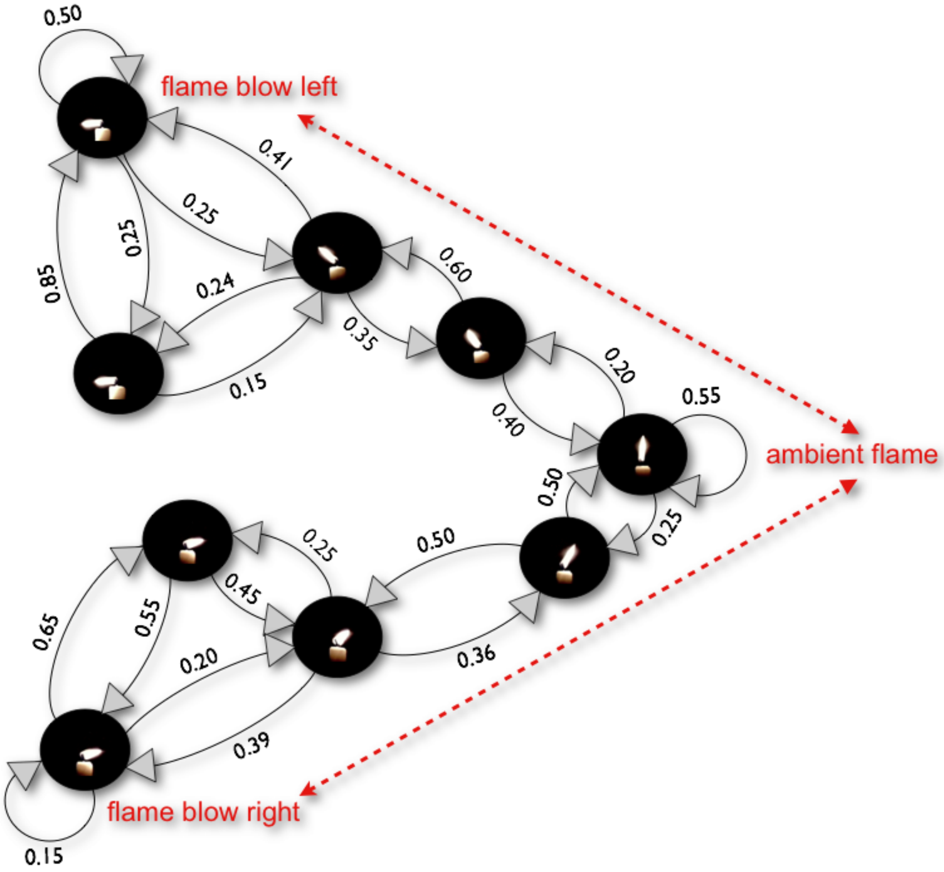


Figure 4.8: **Markov chain:** Example Markov chain model for candle flame data set

going from cluster k to cluster l, and $\sum_l p_{k,l} = 1$ learnt from the training data. We are now able to represent the conditional probability of moving from one cluster to another as $P(C_t|C_{t-1}) = p_{C_{t-1},C_t}$ where C_t is defined as the index for a cluster/state at time t (where t is in unit of frames). This transition matrix is constructed using a *frequentist approach*, whereby the probability of transitioning from cut point group C_{t-1} to C_t is based on how frequent C_{t-1} to C_t transitions occurred given the original data.

The transition probability acts as a weighting, giving higher likelihood to transitions that occur more frequently in the original data. To account for situations where a transition might have zero probability, a nominal value is added to all elements in the transition matrix before normalisation. This allows the model to move between states not represented as transitions in the original sequence, or with a low likelihood.

4.3.3 Generating Novel Sequences

To generate novel motion sequences the procedure is:

1. Find a random start pose configuration in pose space $\mathbf{y}_t^c$.

2. Given $\mathbf{y}_t^c$, find all adjacent cut point neighbours in $\mathbf{Y}_t^c$ as defined in Section 4.3.1, to represent start transition points.

3. Find all associated end transition points $\mathbf{z}_{t,m}^c | m = \{1, ..., Q_t\}$. This gives a set of Q_t possible transitions from the starting point $\mathbf{y}_t^c$ in pose space.

4. Denote the cut point group index that $\mathbf{y}_t^c$ belongs to as C_t.

5. Calculate the likelihood of each transition as:

$$\boldsymbol{\phi}_m = P(C^z_{C_{t+1},m} | C_t) P'(\mathbf{z}^c_{C_{t+1},m}) \tag{4.1}$$

   where $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, .., \boldsymbol{\phi}_{Q_t}\}$.

6. Normalise the likelihoods such that $\sum_{i=1}^{Q_t} \boldsymbol{\phi}_i = 1$.

7. Since a maximum likelihood approach will result in repetitive animations, we randomly select a new start transition point $\mathbf{y}_{t,k}^c$ from $\boldsymbol{\Phi}$ based upon its likelihood as:

$$\arg\min_k \left(\sum_{j=1}^{k} \boldsymbol{\phi}_j \geq r \right) \tag{4.2}$$

   where k is the index of the newly chosen end transition point, $k \in m$, and r is a random number between 0 and 1, $r \in [0, 1]$.

8. If $\mathbf{y}_t^c \neq \mathbf{y}_{t,k}^c$, use linear interpolation to blend $\mathbf{y}_t^c$ to $\mathbf{y}_{t,k}^c$ and reconstruct for rendering:

$$\mathbf{x}_{Lin} = \mu + \mathbf{V}(\alpha(t)\mathbf{y}_t^c + [1 - \alpha(t)]\mathbf{y}_{t,k}^c) \tag{4.3}$$

9. All frames associated to the transition sequence between $\mathbf{y}_{t,k}^c$ and $\mathbf{z}_{t,k}^c$ are reconstructed for rendering as:

$$\mathbf{x}_t = (\mu + \mathbf{V}\mathbf{y}_t) \tag{4.4}$$

10. The process then repeats from step 2 where $\mathbf{y}_{t+1}^c = \mathbf{z}_{t,k}^c$.

4.3.4 Destination Driven Dynamic Programming

When searching for a motion, it is not only important to arrive at a user specified action, but to do so following a motion path that looks natural. In most cases, motion requires sacrificing short term objectives for the longer term goal of producing a smooth and realistic sequence. A naive approach would be to do a depth-first search, exhaustively searching all combinations of cut points. Dynamic Programming [11] and the branch and bound strategy used in [70] are more attractive alternatives.

In this system a destination driven approach combined with dynamic programming is used. By treating the cluster of cut points as states, a trellis is built u steps in the future effectively predicting all possible transitions u levels ahead. The most probable state in level u is derived, and dynamic programming is used to find the shortest path. Though it may take slightly longer to generate a desired motion, the overall result is more realistic. With this approach, potential 'dead-ends' are mitigated, which limited the types of motions that could be generated by the model.

Figure 4.9 demonstrates this approach. The small black dots are cut points, and the big blue circles encompassing them are clusters. From a cut point in cluster 1, a trellis is built 3 levels ahead. A cut point in level 3, cluster 7, is derived as the most probable destination. Dynamic programming is then used to find the shortest path to that destination from the root point.

Rendering speed of approximately 25 frames per second was obtained when using a 3 level trellis $u = 3$. u is selected as the maximum number of trellis levels that allows real-time computation and animation. For all data sets, $u > 3$ resulted in no noticeable improvement in the quality of the synthesised animations, however, greatly reduced rendering speed.

4.4 Multimodal Controller

Thus far, the *Motion Model* randomly generates the most likely set of motion sequences given a starting configuration. To allow real-time control a *Multimodal Controller* is

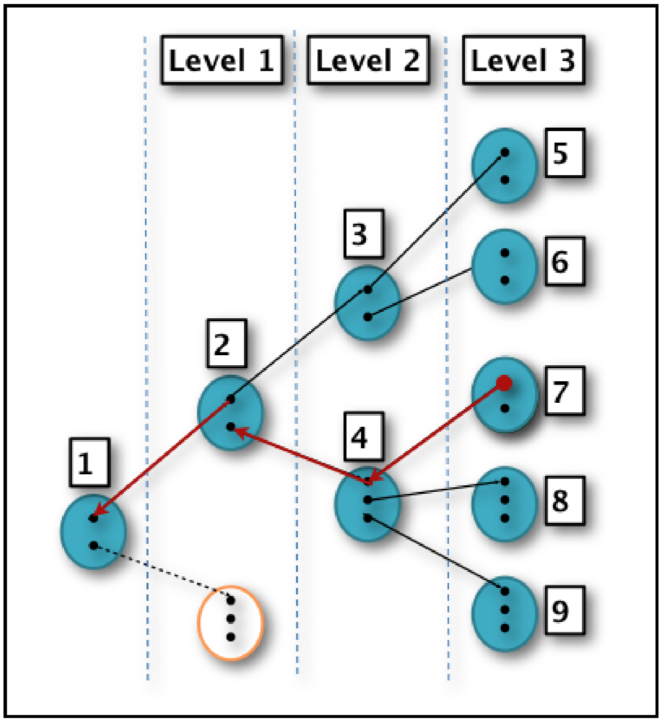


Figure 4.9: **Destination driven dynamic programming:** Example of destination driven dynamic programming

introduced, which uses a conditional probability to map between input space and pose space. This section describes the Projection Mapping method used in controlling the *Motion Model*.

4.4.1 Projection Mapping

The mapping is used when wanting to enable motion control of the generated motion. Firstly, the input space is quantised into an appropriate number of symbols N_s. These symbols are then associated to a set of training examples $E_r = e_i | i = \{1, ..., N_r\}$, where N_r is the number of training examples associated to the r^{th} symbol, and $e_i \in \boldsymbol{\delta}^c$.

The quantisation process is different for each data set and explained in detail in Section 4.5. Taking the plasma beam data for example, as shown in Figure 4.10, the input space is the 2D coordinate-space around the edge of the plasma ball. This space is manually quantised into $N_s = 11$ symbols/sub-regions, relating to the general locations

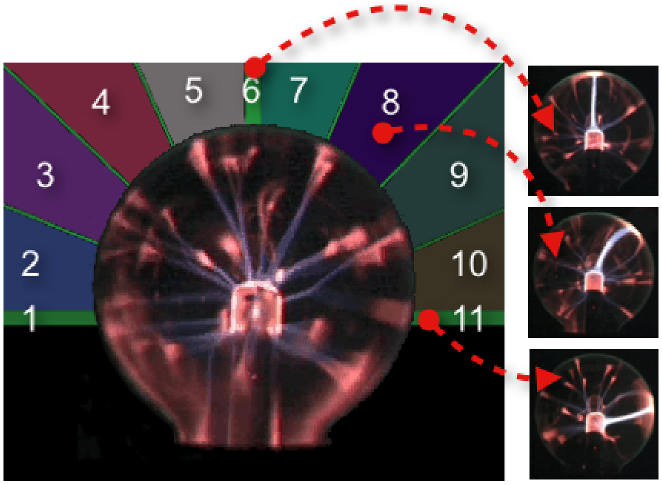


Figure 4.10: **Quantisation process:** Image showing the quantisation of the plasma beam into $N_s = 11$ symbols, relating to the direction the plasma beam can be summoned

the plasma beam can move to.

A conditional probability distribution is built using the training data that maps from the input space to pose space. The r^{th} input symbol is mapped to the q^{th} cut point cluster using $P(C_q|input_r) = p_{q,r}$ (where $input_r \in \{1, ..., Ns\}$), which symbolises the probability of a cut point in cluster q occurring when the user requests the r^{th} symbol.

Given that the r^{th} symbol captured a set E_r of N_r cut point samples, the mapping is computed as:

$$p_{q,r} = \frac{P(input_r, C_q)}{P(input_r)} = \frac{|C_q \cap E_r|}{N_r} \tag{4.5}$$

where $P(input_r) = \frac{N_r}{N_c}$, $P(input_r, C_q) = \frac{|C_q \cap E_r|}{N_c}$ and $\sum_r p_{q,r} = 1$. This is used at run-time to weight the chosen cut points given a user selected input symbol ($input$). As a result, Equation 4.1 is altered to:

$$\phi_m = P(C^z_{C_{t+1},m}|C_t).P'(\mathbf{z}^c_{C_{t+1},m}).W \tag{4.6}$$

where $P(C^z_{C_{t+1},m}|C_t)$ is the probability of transitioning from the current cut point group C_t to the query cut point group C_{t+1}, m, $P'(\mathbf{z}^c_{C_{t+1},m})$ is the likelihood of the respective end transition point, and

$$W = \begin{cases} P(C_i|input) & \text{if } \mathbf{y}_i \in \mathbf{Y}_i^c \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

4.5 Animation/Results

This section presents the results of the MIMiC system demonstrated in three different data formats: MoCap, video, and conversation.

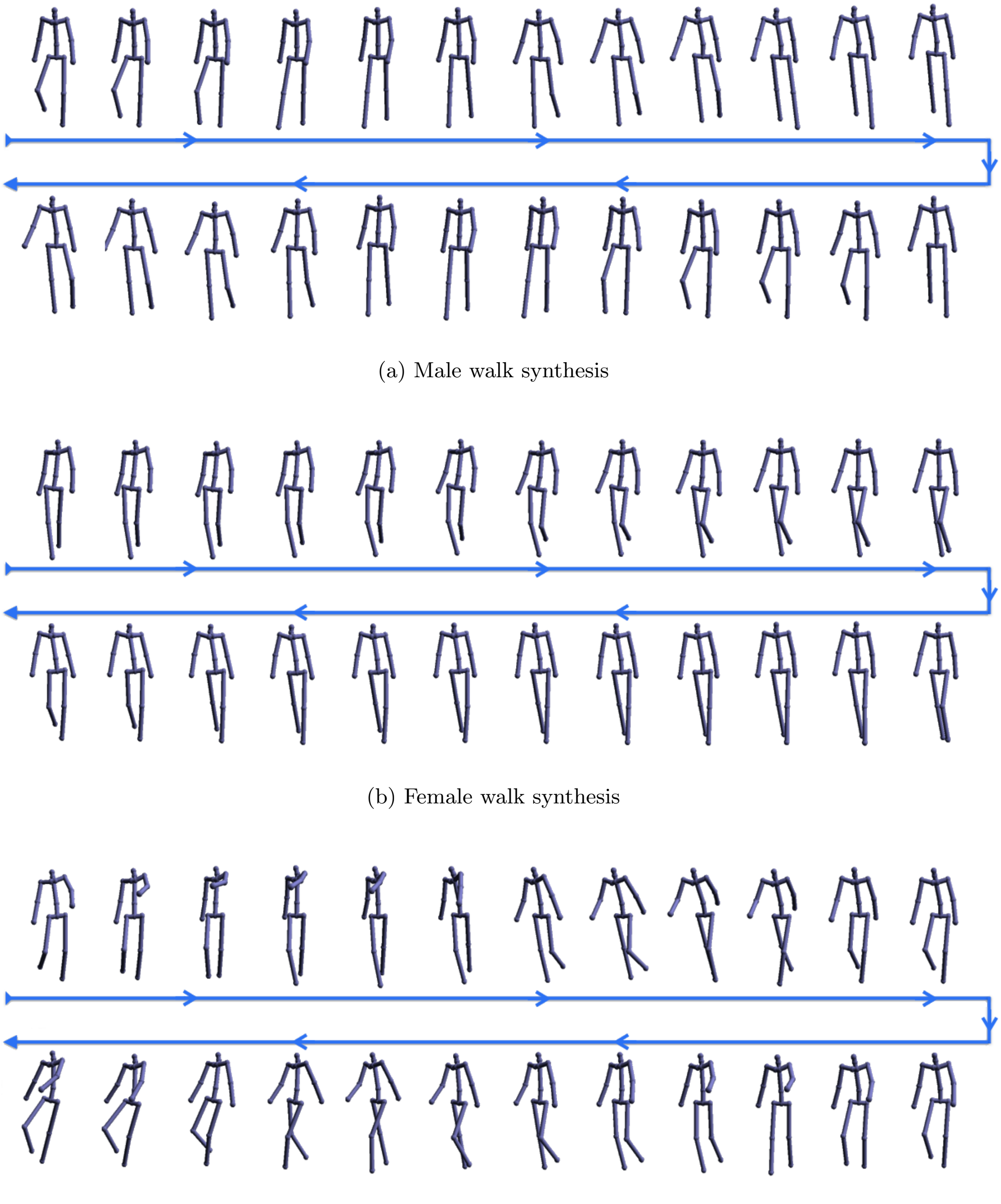
4.5.1 Experiments with MoCap Data

Six motion capture sequences were projected down into their combined lower dimensional eigenspace using the approach detailed in Chapter 3, Section 3.2. This made up a data set of 2884 frames at a reduced 30 dimensions. The six individual motion sequences were of a 'male walk', 'female walk', 'drunk walk', 'skip', 'march', and 'run'. The sequences were captured from the same actor using 36 markers to cover the main joints of the human body. Using our unsupervised segmentation approach, as detailed in Section 4.3.1, 61 k-medoid points were defined, using $\theta = 0.5$ to produce 228 subsequences. In the quantisation process, as explained in Section 4.4.1, $N_s = 6$, relating to the six different types of walks in the data set.

Figure 4.11 and 4.12 presents the results of generating novel sequences of the 6 discrete motion types. Figure 4.13 shows the results of blending between the different types of walks. In this example, the user chooses to animate from a female walk to a drunk walk, then to a male walk, march, run and skip. Frames a, b, c, d, e and f are cut points used to make smooth transitions from one type of walk to another. As suggested by the dotted red lines, these cut points can be used to transition to walks not demonstrated in this example.

4.5.2 Experiments with Video Data

Two video sequences were recorded using a webcam. One was of a candle flame and the other of plasma beams from a plasma ball.

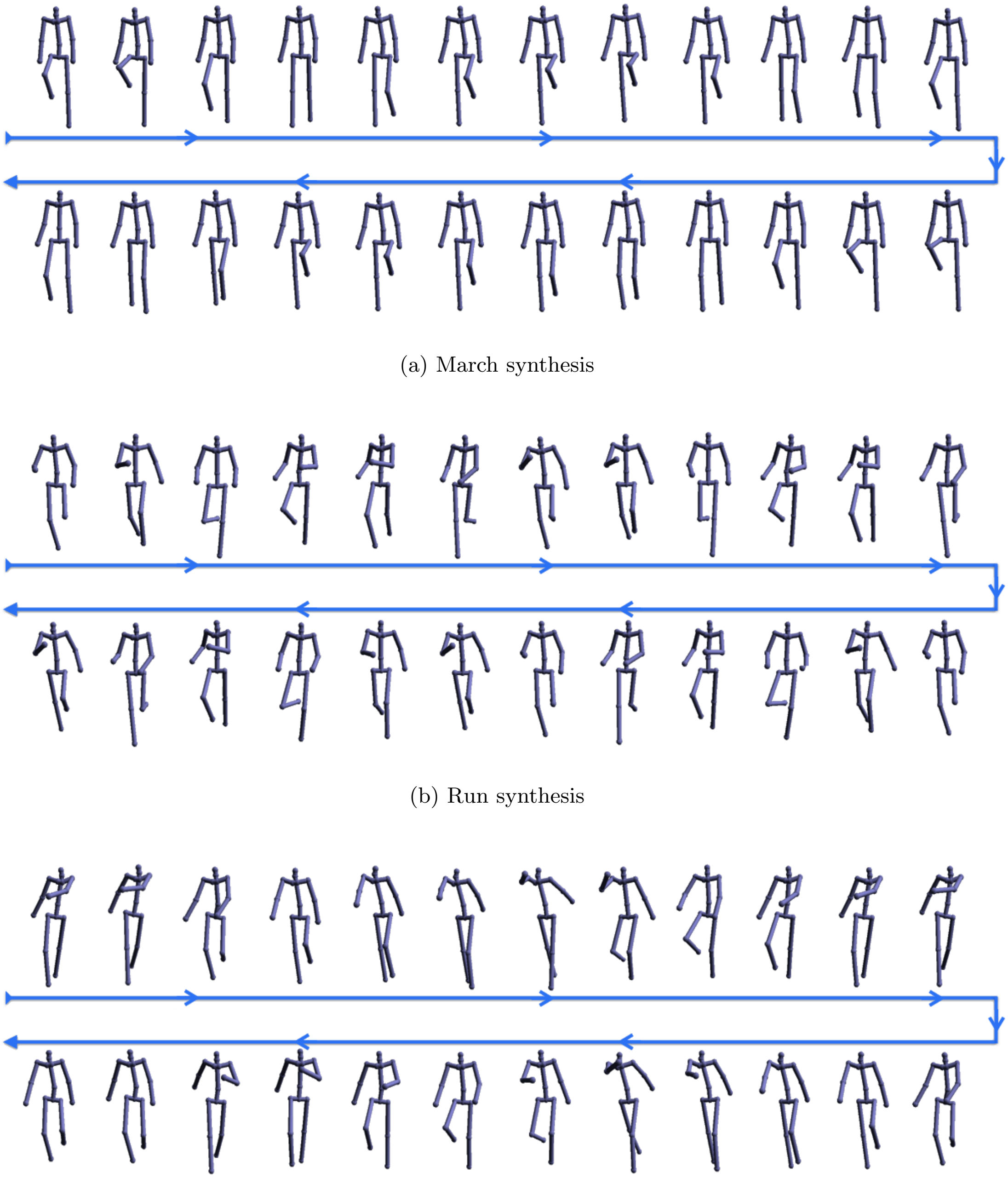


(a) Male walk synthesis

(b) Female walk synthesis

(c) Drunken walk synthesis

Figure 4.11: **MoCap Walk Synthesis:** Image showing synthesis of a male walk (a), female walk (b), and drunken walk (c) using MIMiC.



(a) March synthesis

(b) Run synthesis

(c) Skip synthesis

Figure 4.12: **MoCap Walk Synthesis:** Image showing synthesis of a march (a), run (b), and skip (c) using MIMiC.

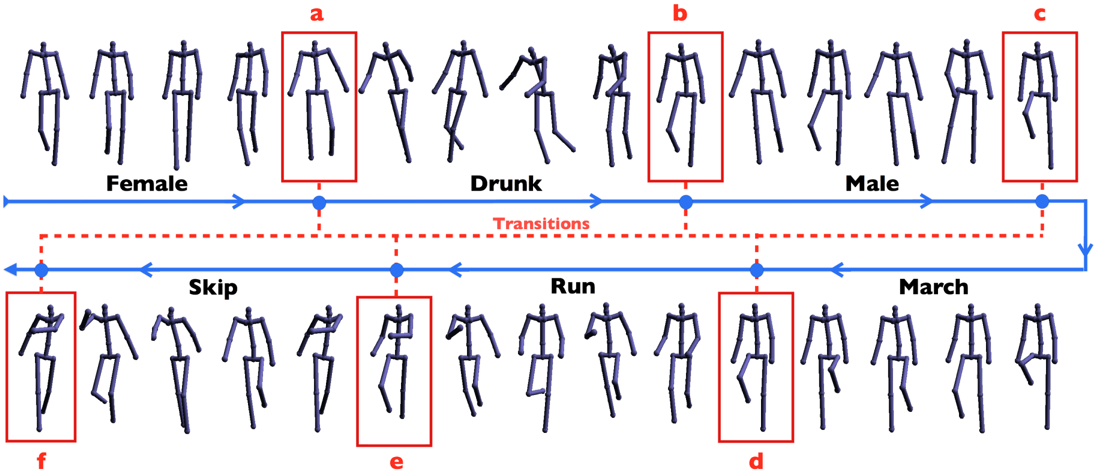
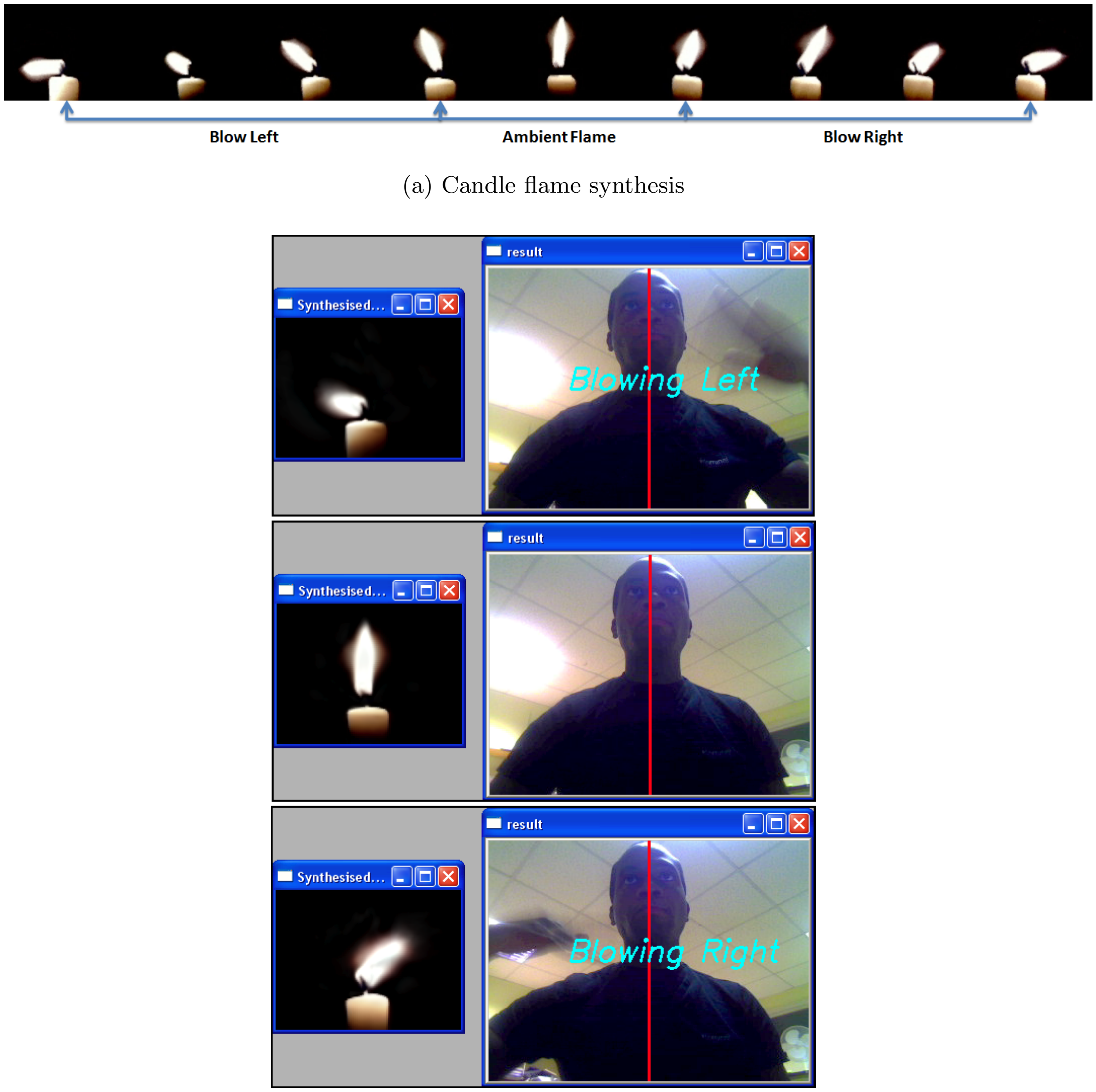


Figure 4.13: **MoCap synthesis:** Image showing synthesis and blending of different types of motion captured walks. The blue arrow indicates the motion trajectory of the motion synthesis. The frames in boxes (a, b, c, d, e and f) are cut points used for transitioning from one type of motion to another.

The candle flame sequence is the same as that used in Chapter 3. The recording was of a candle flame performing 3 different motions, blowing left, blowing right and burning in a stationary position. The dimension reduction process, projected the data down to 42 dimensions. Using our unsupervised segmentation approach, 90 k-medoid points were defined, using $\theta = 0.25$ to produce 309 subsequences. $N_s = 3$ giving the user control over the three discrete states of the candle flame. As shown in Figure 4.14 (a), using MIMiC, the user can control the three discrete states of the candle flame motion and blend between them. If the animation is at a blow right state, it has to travel to the stationary state before a blow left state can be reached, expressed by the transition matrix and determined through dynamic programming. Since the original data is used for synthesis, no *ghosting effect* occurs and the animation looks more realistic. Demonstrated in Figure 4.14 (b), using simple image processing to detect motion, the user directly interacts with the animation by using hand motion to simulate a breeze which affects the direction of the flame in animation.

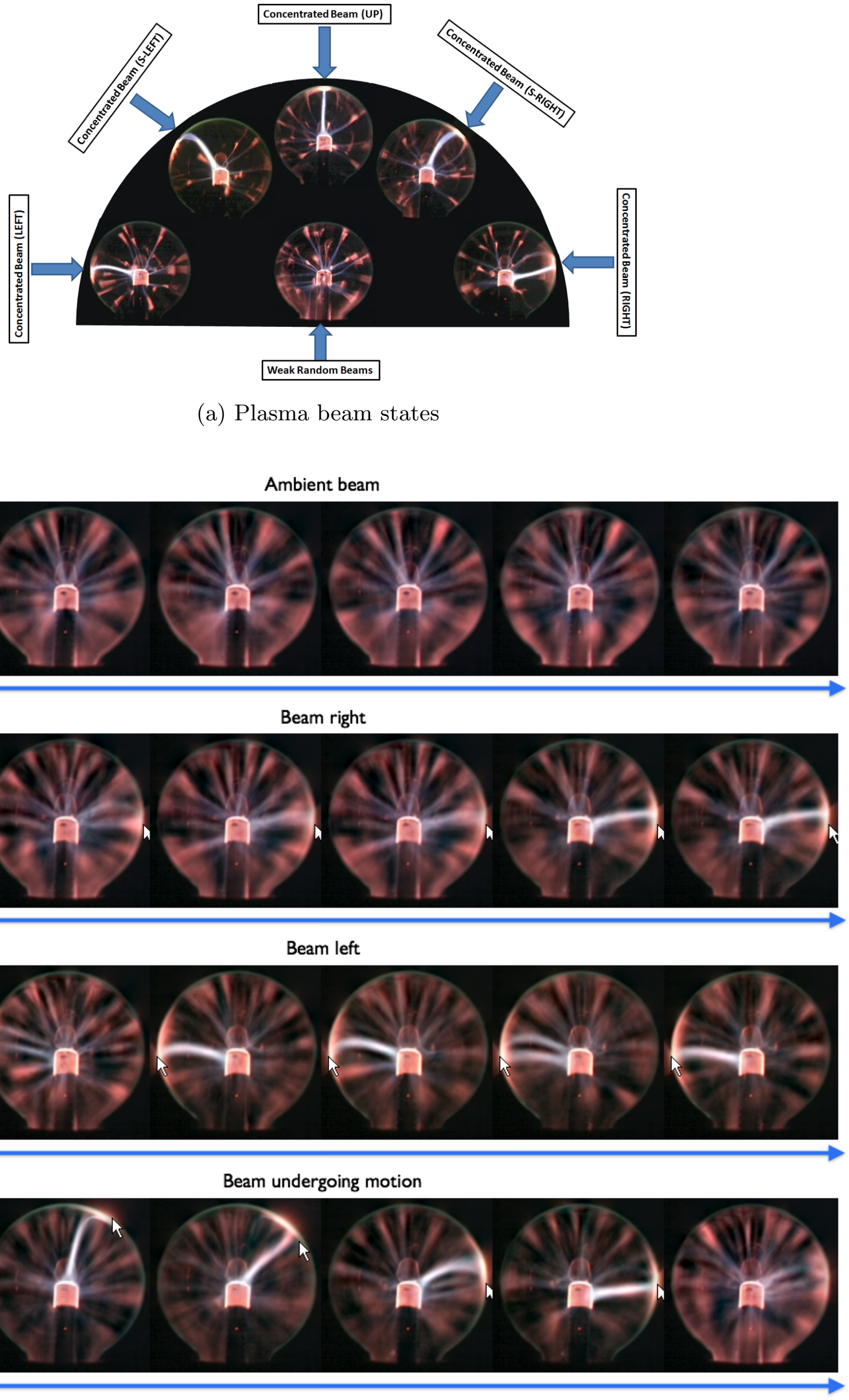
The plasma beam sequence was captured with a webcam (180 × 180 pixels, 15 frames per second). The recording was 3:19 minutes long containing 2985 frames. Dimen-



(a) Candle flame synthesis

(b) Candle flame synthesis (gesture controlled)

Figure 4.14: **Candle flame synthesis:** (a) Image showing candle flame synthesis. Using MIMiC, the user is able to control the three discrete states of the candle flame and blend between them. To transition from a flame blow left state to a blow right state, the system will perform a transition to a stationary flame state first resulting in a better looking transition. (b) Screenshot of control of candle flame using hand gestures. The gestures simulate a breeze which effects the direction of the flame in the animation.



(a) Plasma beam states

(b) Plasma beam synthesis (mouse cursor controlled)

Figure 4.15: **Plasma beam synthesis:** Image showing plasma beam states (a) and mouse controlled synthesis (b).

sion reduction projected this data set down to 100 dimensions, and the unsupervised segmentation algorithm defined 53 k-medoid points, using $\theta = 0.6$ to produce 230 subsequences. Figure 4.15 (a) shows the varying states of the plasma beam. The plasma beam sequence has more varying movement than the candle flame. It produces motion ranging from multiple random plasma beams to a concentrated beam from a point of contact anywhere around the edge of the ball. As a result, the modelled plasma beam offers more varying degrees of control. The plasma beam motion is divided into 11 discrete states around the edges of the plasma ball. Using a mouse cursor or touch screen, the user can control the movement of the plasma beam, as shown in Figure 4.15 (b). Synthesis is demonstrated on ambient beam, beam right, beam left, and beam undergoing motion from top to right. Again, since the original data is used for synthesis, no *ghosting effect* occurs.

4.5.3 Experiments with Conversation Data

As shown in Figure 4.16 (a), two people (person A and B) conversing with each other were recorded using two SD (Standard Definition) cameras (720 × 576 pixels, 25 frames per second) and a microphone (48kHz). Figure 4.16 (b) shows their configurations with respect to themselves, the cameras, and the microphone. They sat face to face at a comfortable distance apart. The frontal view of each face was captured whilst they conversed for 12 minutes. They spoke in fluent english and considered themselves friends.

The data was analysed and one of the subjects was chosen to be the expressive *listener* whilst the other was deemed the *speaker*. Periods when the *listener* is clearly engaged in listening to the *speaker* with no co-occurring speech were extracted. This produced 10 audio-visual fragments which were combined to produce a total of 2:30 minutes of data.

As demonstrated in Figure 4.16 (c), the facial features of the listener, including head pose, were tracked using a Linear Predictor tracker [93]. 44 2D points were used to cover the contour of the face including the eye pupils. When processed, this produced 55 k-medoid points and 146 subsequences using $\theta = 0.03$, which we reduced to 50

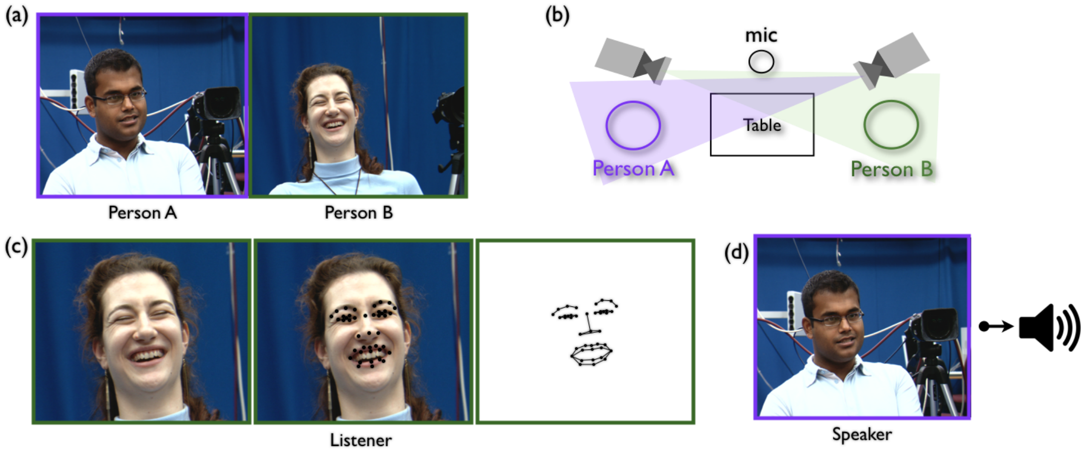


Figure 4.16: **Extracting listener and speaker:** (a) Video and audio data of two people in a natural conversation. (b) Configuration of the two conversers in the room. Two cameras recorded the frontal face view of each person, and a single microphone recorded the audio. (c) An expressive listener was chosen, and an LP tracker [93] was used to track the contours of their facial features. (d) Speaker is chosen and the audio stream of the speaker is extracted.

dimensions using PCA. As shown in Figure 4.17, the movements of these 2D points are dynamically generated from MIMiC in real-time. The audio stimulus uses the conditional probability to derive various visual responses based on its content. The most prominent visual responses are head nods, although other expressions like smiles, eye brow lifts and blinks are generated when appropriate.

The audio stream of the speaker, as shown in Figure 4.16 (d), is represented using 12 Mel-Frequency Cepstral Coefficients (MFCCs) and a single energy feature of the standard HTK setup, a configuration commonly used in speech analysis and recognition [79]. The frame rate of 100 frames per second was selected with 50% overlap, i.e., the window size is $20ms$ and the step size $10ms$. $N_s = 25$ symbols/classes of the speaker's MFCC is used as the input space. The extraction of these classes are automatic using the k-means algorithm. Here, N_s is chosen experimentally to represent an even distribution of the MFCCs. The conditional probability, as explained in Section 4.4.1, is then learnt that maps MFCC input features to pose space to map the audio features to the animation.

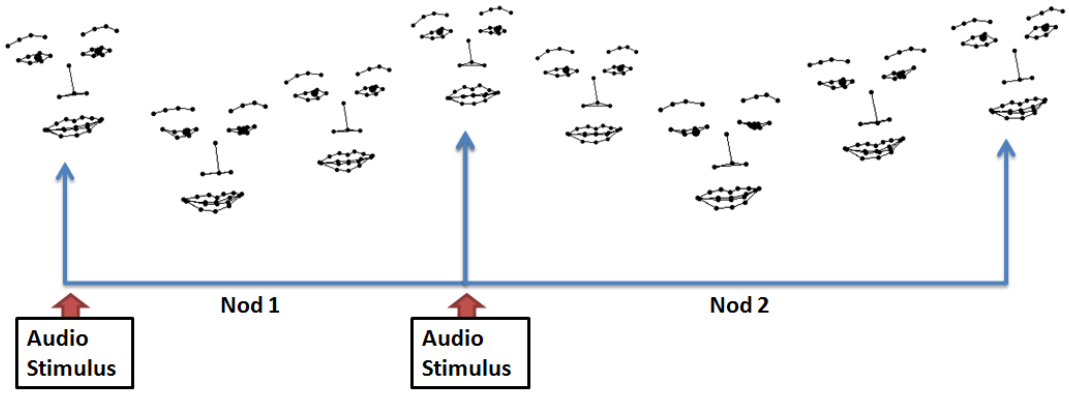


Figure 4.17: **Synthetic listener:** Image showing synthesis of nods as responses to audio stimulus

For testing, another set of audio sequences were captured from the same speaker in a casual conversation. 15 speech fragments were selected from the conversation totalling 2:31 minutes. Using the projection mapping from audio features to *pose space*, these speech fragments generated a synthetic listener with plausible visual responses.

To validate results, 14 people were asked to listen to the 15 test audio segments and to score between 1 and 10 how well the visual model responded to the audio as a synthetic listener in the conversation. They were unaware that approximately half of the visual responses to the audio segments were playing randomly regardless of the audio input, whilst the other half were generated from the audio input to the model. The results are listed in Table 4.1. We normalised each person's score and took the average for both audio-model generation and random play. As shown in the fourth column of Table 4.1 entitled '$\frac{Model}{Random}$', 11 out of 14 generated a score greater than or equal to 1, showing preference to the visual responses generated by the audio input than by the random play.

A statistical significance pair t-test was also performed on the scores on Table 4.1, to evaluate whether the distinctions between the scores are notable with regards to their standard deviation. With $p\text{-}value = 0.0196$ ($p < 0.05$), the null hypothesis is rejected such that there is sufficient evidence to suggest that there is a difference in means across both scores when also considering their standard deviation.

Person	**Audio-Model**	**Random Play**	$\frac{\textbf{Model}}{\textbf{Random}}$
1	0.66	0.32	**2.1**
2	0.56	0.43	**1.3**
3	0.86	0.11	**7.8**
4	0.45	0.56	0.8
5	0.45	0.56	0.8
6	0.9	0.1	**9**
7	0.27	0.7	0.4
8	0.78	0.18	**4.3**
9	0.55	0.43	**1.3**
10	0.78	0.2	**3.9**
11	0.65	0.33	**2**
12	0.7	0.3	**2.3**
13	0.5	0.5	**1**
14	0.57	0.42	**1.3**
Aver.	0.62	0.37	
Std. Dev.	0.18	0.18	

Table 4.1: **Scores for visual responsiveness:** Scores for visual responses to audio. Column 1 is the numerical index of people giving scores. Column 2 and 3 are the normalised and averaged scores for visual responses that are audio driven and randomly playing respectfully. Column 4 is the audio driven scores divided by the random play scores

Person	**Replay**	**Random Play**
Aver.	0.57	0.45
Std. Dev.	0.3	0.33

Table 4.2: **Average and standard deviation:** Average and standard deviation of scores for visual responses to audio based on replay of original data and random play

Although the majority could tell the difference, the margins of success are not considerably high producing an average of 0.62. Several assumptions may be drawn from this. As nods are the most effective non-verbal response of an engaged listener, random nods may provide an acceptable response to a *speaker*. To try to validate these tests, the same 14 people were asked to repeat the test but this time on the 10 audio segments used in training the model. 5 out of 10 of the audio segments were randomly played visual responses and the other 5 were replays of the original audio-visual pairing. Results in Table 4.2 show that for a baseline test on ground truth data, where we know a direct correlation exists between the audio signal and the visual response, the participants provide very similar levels of scoring. This indicates that our animations are very nearly as convincing as a real listener in terms of the responses provided to audio data.

4.6 Summary

MIMiC can generate novel motion sequences of various formats, giving a user real-time interactive and multimodal control of animations. Unlike a generative model, MIMiC can be applied to more complex motion types. Since the original data is used in synthesis as opposed to a generalisation, the quality and realism of the animation is assured, eliminating the possibility of *ghosting efftects*. The novel motion segmentation approach allows for rapid derivation of transition points, even for large data sets. MIMiC was also successfully extended to HCI, modelling conversational cues for a synthetic listener, deriving appropriate responses using audio features.

Building on the idea of modelling conversational dynamics between individuals, the next chapter explores a method for modelling contextual information in natural conversation. This is based on analysing social signals such as eye gaze direction, nodding, and laughing, in the context of conversation interest.

Chapter 5

SDM: Social Dynamics Model

The MIMiC system in Chapter 4, provides multimodal control of motion data. The approach was further extended to HCI, creating a synthetic listener capable of responding non-verbally (nodding) to appropriate audio speech features. However, the synthetic listener is only capable of a single social response, and since contextual information is not incorporated, this artificial listener is only capable of functioning in the simplest of social encounters, i.e. [*speaking-to-me* ⇒ *respond*], [*not-speaking-to-me* ⇒ *do-not-respond*].

To allow this approach to operate in more diverse social contexts, a means of parameterising the multimodal social dynamics between the speaker and the listener is required. In this chapter, such an approach is presented. It introduces a novel method for modelling social dynamics governed by the exchanges of non-verbal cues between people. Unlike other social models that rely on intangible psychological observation, this approach uses tangible rules governed by the data to discern distinct trends and characteristics.

Apriori association rule mining is used to deduce frequently occurring patterns of social trends between a speaker and a listener in both *interested* and *not interested* social scenarios. The *confidence* values from rules are used to build a *Social Dynamic Model* (SDM). In this chapter, this model is demonstrated on classification and visualisation. By visualising the rule generated in the SDM, distinct social trends between an *interested* and *not interested* listener in a conversation can be analysed. Results show that

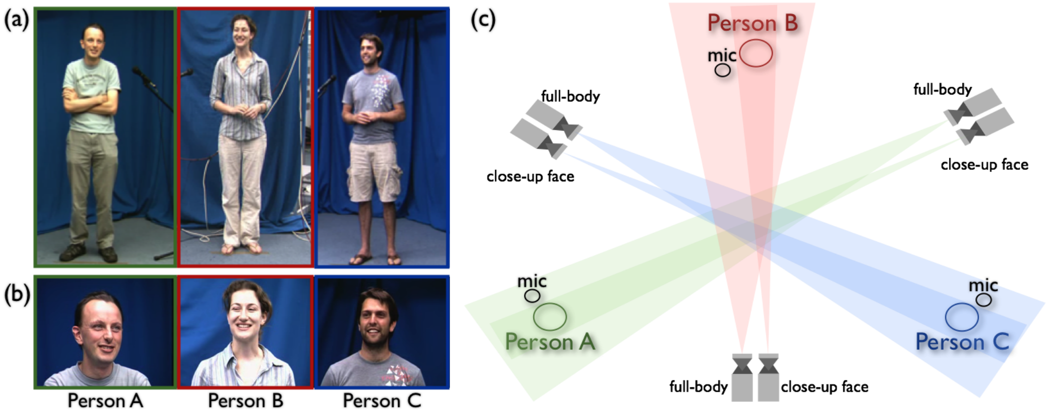


Figure 5.1: **Data set:** (a) Image showing full-body view of recorded video data of three individuals having a conversation. (b) Image showing close-up face view. (c) Diagram showing the configuration of cameras, microphones (mic) and conversers. We refer to the three individuals in the conversation as person A, B, and C.

these distinctions can be applied generally and used to accurately predict conversation interest. In Chapter 6, this model is further extended to allow for autonomous control of socially interactive avatars.

This chapter is divided into the following sections. Section 5.1 presents the conversation analysis where experiments are conducted to try and deduce distinct rules that dictate the social dynamics of people in a conversation. Section 5.2 provides the visualisation and interpretation of SDM, and the remainder of the chapter presents an evaluation and conclusion.

5.1 Conversation Analysis

5.1.1 Video Data Set for SDM

The data set consists of approximately 30 minutes of video and audio recording (43000 frames) of the full-body frontal view (516×340, 25 frames per second, 48kHz) and the close-up frontal face view (720×576, 25 frames per second, 48kHz) of 3 individuals having a conversation with each other. An image of the full-body recording for each person is shown in Figure 5.1 (a), and the face recording in Figure 5.1 (b). We refer to the 3 individuals as person A, B, and C. Each person remained in a stationary position relative to the cameras as shown in Figure 5.1 (c).

5.1.2 Social Behaviour Experiment

Prior to capture, each person was given a questionnaire and asked to score from 1 to 3 their interest (where 1 is low interest, 2 is moderate interest, and 3 is high interest) on a given set of *book genres*, *film genres* and *music genres*. If the subjects had no knowledge or strong opinions about the topic, they would score it 0. They were also given specific questions like: *favourite sports*, *language(s) spoken fluently*, *favourite music concerts*, *favourite theatre show* etc. Their questionnaires were analysed to choose topics for conversation that would lead to the following 4 generic scenarios:

1. All highly interested in the topic
2. Two people highly interested in the topic, one person has a low interest
3. One person highly interested in the topic, two people have low interest
4. All have low interest in the topic

These 4 generic scenarios where derived from 8 topics of conversation as detailed in Table 5.1. The sixth column of Table 5.1 shows the limited duration of each topic, chosen to suite the scenario. A projector displayed the topic of conversation for discussion, and a quiet bell would ring to make the subjects aware of the change in topic. The subjects were unaware of the nature of the experiment, and were simply asked to discuss the topic displayed on the screen.

The aim of this experiment was to observe the social dynamics between the three people in scenarios when *interested* or *not interested* in the topics. The next step is to quantise their social signals (from the data) in a form that is suitable for data mining in order to obtain rules governing social behaviour.

Scenario	A	B	C	Topic	Period
1	3	3	3	Classical Music	5 min
2	3	3	1	Adventure Novels	5 min
3	3	1	3	Philosophy Novels	5 min
4	1	3	3	Rock Music	5 min
5	3	1	1	Sailing (Spoken in French)	2.5 min
6	1	3	1	Triathlon/Les Miserables (Spoken in Afrikaans)	2.5 min
7	1	1	3	Radio Head Concert	2.5 min
8	1	1	1	Horror Novels	1.5 min

Table 5.1: **Different social scenarios:** Table showing 8 different social scenarios dictated by the topic of conversation. The three people are referred to as person **A**, **B**, and **C**. The numbers indicate their interest in the topics where 3 is a high interest and 1 is a low interest

5.1.3 Semi-Supervised Social Signal Annotating

Pentland [97] proposed measuring non-linguistic social signals using four main observations: *activity level*, *engagement*, *emphasis* and *mirroring*. Using this as a base, 7 social signals in the conversation are observed: *Voiced*, *Talking*, *Laughing*, *Head Shake*, *Head Nod*, *Activity Measure*, and *Gaze Direction*.

We use a variety of techniques to derive each annotation.

1. **Voiced[V]:** The audio stream is represented using 12 MFCCs (Mel-Frequency Cepstral Coefficients) and a single energy feature of the standard HTK setup [79]. This is the same audio configuration used in the MIMiC *conversation data* experiment in Chapter 4, Section 4.5.3. For each person, a few voiced segments were annotated and a Mahalanobis distance measure was used to derive a correlation between the voiced and non-voiced regions.

2. **Talking[T]:** With the voiced segments annotated, it was a simple process of annotating the voiced segments which were talking. This was done by hand.

3. **Laughing[L]:** The Viola-Jones face detector [130] was used to segment the face

region in each frame. The lip region was localised by cropping the lower-centre region of the face. An AdaBoost classifier was then trained for laughing and used to annotate the remaining data.

4. **Head Shake[S]:** The Viola-Jones face detector was used to determine the movement of the face. A Fast Fourier Transform (FFT) was used to identify high frequency movement along the x-axis

5. **Nod[N]:** Similar to head shakes, an FFT was used to identify high frequency movement along the y-axis.

6. **Activity Measure[A]:** The torso region of the full body video was segmented using colour and the mean-scaled standard deviation of velocity was measured. The leg and head regions are ignored because, there was minimal leg movement (subjects remained stationary), and since we are more interested in gesture activity, changes in head posture/gaze would bias the activity measure.

7. **Gaze Direction[G]:** The eye pupils and the corners of the eyes were tracked using a Linear Predictor tracker [93]. The positions of the corner of the eyes were normalised to 0 and 1, and the resulting position of the eye pupil within this region was used to determine if the person was gazing left [GL], right [GR] or centre [GC].

5.1.4 Social Signal Labels

For each person, 9 social signals are observed and given labels, where labels $1-6$ are as numbered in Section 5.1.3, and 7, which is the *gaze direction* signal, consists of 3 parts, *gaze left*, *gaze right*, and *gaze centre*, which associate to labels $7-9$ respectively. As shown in Table 5.2, this produces N_T sets of person specific social signal labels (where N_T is the total number of frames) of 27 dimensions, where $1-9$ is for person A, $10-18$ for person B and $19-27$ for person C.

To allow data mining to generalise about the exchanges in social signals between the 3 participants, two items are required. Firstly, the exchanges in social signal between the

Per.	1	2	3	4	5	6	7	8	9
A	[V]	[T]	[L]	[S]	[N]	[A]	[GL]	[GR]	[GC]
Per.	10	11	12	13	14	15	16	17	18
B	[V]	[T]	[L]	[S]	[N]	[A]	[GL]	[GR]	[GC]
Per.	19	20	21	22	23	24	25	26	27
C	[V]	[T]	[L]	[S]	[N]	[A]	[GL]	[GR]	[GC]

Table 5.2: **Social signal labels:** Set of social signal labels in each frame for person A, B, C. $1-9$ is for person A, $10-18$ for person B and $19-27$ for person C

speaker and listener, which will be referred to as *instances*, need to be defined. Secondly, *person independent* social signal labels are needed to generalise these *instances*.

For each frame, a set of *instances* are defined. A single *instance* is the list of social signals exchanged between 1 listener and 1 speaker. In each frame, the speaker is determined by the person annotated as *talking* [T]. A speaker is not necessarily present in every frame. However, in the majority of cases, there are 2 listeners and 1 speaker, resulting in two *instances*, one for each listener's social exchanges with the speaker. In cases when there is co-occuring speech between 2 or more people (which tends to take place when the listener turn-takes from the speaker), separate *instances* are defined for each speaker in turn, such that each is given the opportunity to be both the speaker and listener. This can result in multiple *instances* for a given frame (a maximum of 6 *instances*).

Person independent social signal labels are used to represent the general occurrence of the social signal regardless of the person. This is shown in Table 5.3. The top row (Listener), are labels for the *person-independent listener*, the second row (Speaker), are labels for the *person-independent speaker*, and the bottom row are their respective signals. With regards to the speaker, *voiced* [V] and *talking* [T] social signals are ignored since the speaker is guaranteed to be voiced and talking.

Using these generalised labels, two sets of instances for *interested* and *not interested* scenarios are defined separately as $\mathcal{F}_{(int)}$ and $\mathcal{F}_{(not)}$, such that $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^{N_I}$ where $\mathbf{f}_i$ is

Listener	PI-1	PI-2	PI-3	PI-4	PI-5	PI-6	PI-7	PI-8	PI-9
Speaker	-	-	PI-10	PI-11	PI-12	PI-13	PI-14	PI-15	PI-16
Signal	[V]	[T]	[L]	[S]	[N]	[A]	[GL]	[GR]	[GC]

Table 5.3: **Generalised social signal labels:** Set of generalised social labels in each *instance*. The top row are labels for the *person-independent listener*, the second row are labels for the *person-independent speaker*, and the bottom row are their respective signals

a 16 dimensional binary vector, and N_I is the total number of instances.

5.1.5 Data Mining for Frequent and Distinctive Social Trends

This framework is driven by the speaker. At any given *instance*, there is one speaker and one listener. We are interested in the combination of social signals a listener performs given a speaker's social behaviour when the listener is *interested* and *not interested* in the conversation. Manually observing all combinations of listener and speaker behaviours in such a large data set would be virtually impossible. A solution would be to make some common sense prior assumptions of expected trends (i.e. an interested listener would gaze more at the speaker than when they are not interested) and focus primarily on these assumptions. However, there is no way of proving or disproving such assumptions, and, there is a large list to chose from.

The idea is to employ a data driven approach to learn such rules. A novel approach to deriving social dynamics and trends between the subjects based on data mining [1] is employed. Data mining allows for large data sets to be processed to identify the reoccurring patterns within the data in an efficient manner. In this framework, *Apriori Association rule* [1, 2] mining is used. Formally developed for supermarkets to analyse millions of customer's shopping trends, the aim is to find *association rules* between a speaker and listener that indicate *interested* and *not interested* from the multitude of possible rules that could exist.

An association rule is a relationship of the form $\{R_i^A\} \Rightarrow R_i^C$ where R_i^A is a set of social signals of the speaker, and R_i^C a social signal of the listener. $R_i^A = \{r_{i,1}^A, ..., r_{i,|R_i^A|}^A\}$ is the antecedent where r_i^A denotes a speaker's social signal label, and $R_i^C = \{r_{i,1}^C, ..., r_{i,|R_i^C|}^C\}$ the consequence where r_i^C is a listener's social signal label. We will refer to r_i^A and r_i^C as itemsets of the data to be mined.

The object of Apriori mining is to efficiently find frequent itemsets of association rules. An example would be, if $R_1^A = \{\text{PI-10,PI-12}\}$, and $R_1^C = \{\text{PI-3}\}$ as defined in Table 5.3, then, $\{R_1^A\} \Rightarrow R_1^C$ would imply 'when the speaker laughs and nods, an *interested* listener is very likely to also laugh'. The belief of each rule is measured by a *support* and *confidence* value. The *support* measures the statistical significance of a rule, it is the probability that a transaction contains itemset R_i^A.

$$sup(\{R_i^A\} \Rightarrow R_i^C) = sup(\{R_i^A\} \cup R_i^C) \tag{5.1}$$

The *confidence* is the number of occurrences in which the rule is correct, relative to the number of cases in which it is applicable.

$$conf = \frac{sup(\{R_i^A\} \cup R_i^C)}{sup(R_i^A)} * 100 \tag{5.2}$$

Mining can return all association rules which meet a prior support and confidence value for user specified itemsets. This alters the quantity and quality of the rules returned. The higher the support, the more frequent the rule in the database. The higher the confidence, the stronger the rule, i.e. the more often the antecedent results in the consequence. For these experiments, only rules with *support* values less than 0.5 are discarded.

The Apriori algorithm uses a *bottom up* approach to efficiently determine the frequently occurring itemsets that meet the support value criteria. The frequent subsets are extended by one item each time, and the infrequent subsets are pruned. As such, itemsets of length g are used to explore itemsets of length $g + 1$. This makes the assumption that, if a subset of one or more items does not meet the minimum *support* value, the same subset extended by one item will also not meet the minimum *support* value. This process continues until no new subset can meet the support value criteria. The successful candidates are then used to search the database and generate association rules that meet the confidence value criteria (see Equation 5.2).

Apriori association mining is applied to $\mathcal{F}_{(int)}$ and $\mathcal{F}_{(not)}$ (as defined in Section 5.1.4) separately, to derive frequently occurring association rules. Traditionally, apriori association rule mining looks for a combination of symbols that occur simultaneously, however, a listener's social behaviour is always a response to the speaker's social signals, hence, co-occurance is unlikely. To account for this, *temporal bagging* within a set temporal window is used to enforce a temporal coherence between features. Given a speaker's social signal, the listener's social behaviour $s = 10$ frames in the future (approx $\frac{1}{2}$ a second) is observed.

5.2 Visualising and Interpreting SDM

The SDM allows visualisation of multimodal trends in social interaction between a speaker and listener in a conversation. Using the mined *confidence* values, the conditional probability of the listener's social responses, given the speaker's social signals, can be visualised to identify social trends without needing to rely on observation alone.

To avoid over complicating the diagram with the numerous combinations of association rules, only association rules with single antecedents (i.e. $|R_i^A| = 1$) are shown, whereby the likelihood of a listener's social response is derived by a single speaker's social signal. The more complex rules are still kept in the model, however, the simpler rules are used for visualisation to discern prominent trends.

The visual-SDM is made up of two components: a **skeleton** and a set of **pentagon rings**. The skeleton consists of 7 black lines that collectively meet at a central intersect. Each line represents a different speaker's social signal and are configured as shown in Figure 5.2 (a) (the annotations are as detailed in Section 5.1.3). As mentioned earlier, with regards to the speaker, *voiced* [V] and *talking* [T] social signals are ignored since the speaker is guaranteed to be voiced and talking. To add clarity to the gaze labels, instead of [GL], [GR] and [GC], we use [G-S], [G-OL], [G-N] representing *gazing at speaker*, *gazing at another listener* and *gazing at no one*, respectively.

The second component is a set of 9 pentagon rings. Each ring represents a different listener's social signal. As presented in Figure 5.2 (b), the individual rings are coded

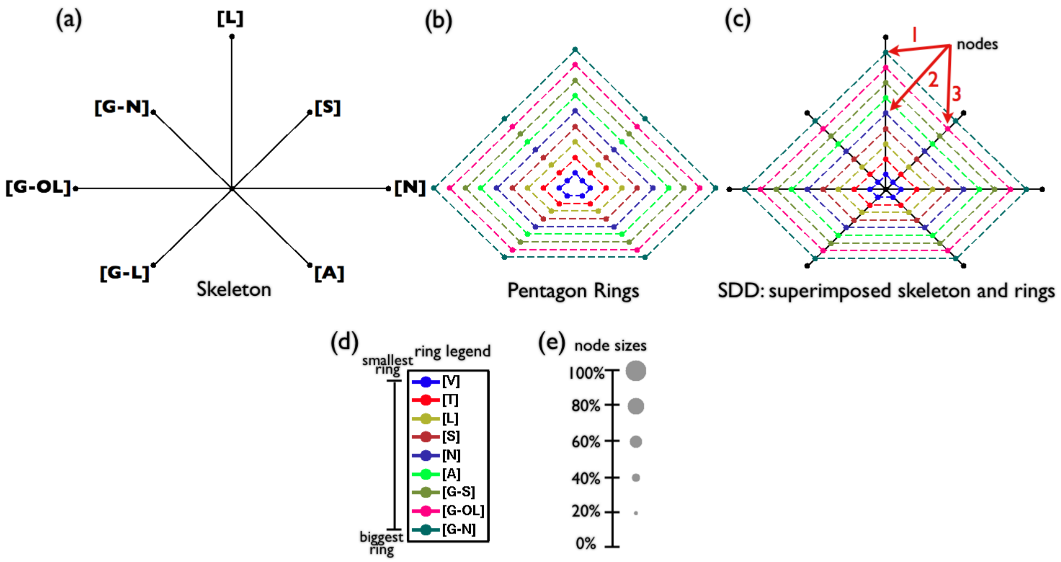


Figure 5.2: **Visual-SDM structure:** (a) The skeleton of SDM. Consists of 7 black lines that are attached to a central intersection. Each line represents a different speaker's social signal (b) 9 pentagon rings where each ring represents a listener's social response. The individual rings are coded by colour and size. (c) SDM is made up of the rings superimposed on the skeleton. The points where the rings intersect the skeleton are known as nodes and infer a listener's social response given a speaker's social signal. A few are indicated by the three red arrows (arrows 1, 2, 3). Arrow 1 is pointing at node [L] ⇒ [G-N], arrow 2 at node [L] ⇒ [N], and arrow at node [S] ⇒ [G-OL]. The idea is that the nodes will vary in size reflecting the respective mined *confidence* value. (d) The legend for the pentagon rings. (e) Shows the relationship of *confidences* value to node sizes.

by colour and size. The ring legends are detailed in Figure 5.2 (d). Shown in Figure 5.2 (c), the superimposed skeleton and pentagon rings make up the visual-SDM. The points where the rings intersect the skeleton infer the occurrence of a listener's social response given a speaker's social signal. We refer to these points as nodes, three of which are indicated by the red arrows (arrows 1, 2, and 3) in Figure 5.2 (c). Arrow 1 is pointing at node [L] ⇒ [G-N], denoting when speaker laughs, the listener gazes at no one. Arrow 2 is point at node [L] ⇒ [N] (when speaker laughs, the listener nods), and arrow 3 at node [S] ⇒ [G-OL] (when speaker head shakes, the listener gazes at the other listener). These nodes can vary in diameter, reflecting the size of the mined *confidence* value given the rule. A set of example node sizes are presented in Figure 5.2

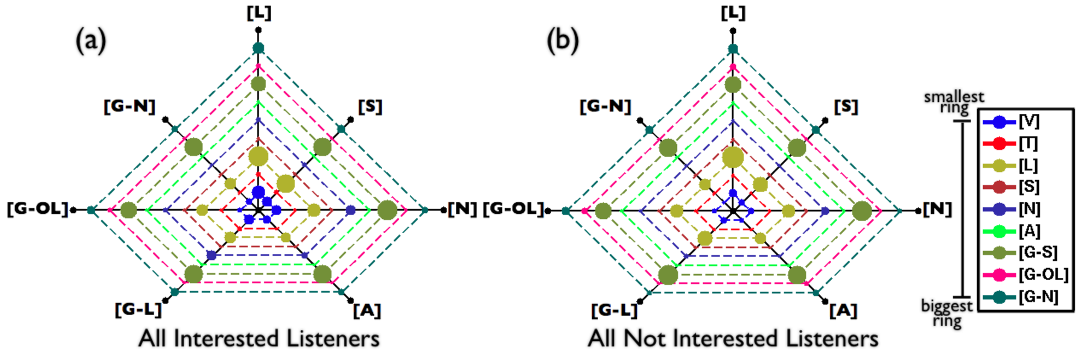


Figure 5.3: **Generated SDMs:** (a) SDM generated from the mined *interested* listener's confidence values. (b) SDM generated from the mined *not interested* listener's confidence values.

(e). Using this structure, prominent rules can be efficiently visualised when comparing social scenarios, simplify a potentially complex set of social behavioural information.

5.3 Conversation Interest Evaluation

5.3.1 Identifying Distinct Trends

To identify trends in social behaviour between a listener and speaker in an *interested* and *not interested* scenario, data mining is performed separately on the derived *instances* of the *interested* and *not interested* data sets.

The *interested* scenario produced 34878 *instances*, whereby, data mining extracted 357 rules in total, 63 of which had single antecedents (i.e. $|R_i^A| = 1$), 153 with two antecedent, 133 with three antecedents, and 8 with four antecedents. The *not interested* scenario produced 33084 *instances*. In this scenario, data mining extracted 396 rules, consisting of 63 rules with single antecedents, 162 with two antecedents, 162 with three antecedents, and 9 with four antecedents. Such complex rules (up to 4 dimensions/antecedents) would be impossible to derive any other way than analytically. Using the *confidence* values derived from these rules, two SDM were built as shown in Figure 5.3 (a) and 5.3 (b). Figure 5.3 (a) is the SDM of a speaker given an *interested* listener and Figure 5.3 (b) is the SDM of a speaker given a *not interested* listener.

By observing both diagrams, the similarities they share are instantly noticeable. All nodes on the third pentagon from the top (third biggest ring), representing the listener's social response [G-S] (gazing at speaker), are large in all instances of the speaker's social signals in both diagrams. A similar trend exists (with minor variations) in nodes on the third pentagon from the bottom (third smallest ring), representing the listener's social response [L] (laughing). From this observation we can discern, contrary to some social interaction studies, that neither a constant gaze at speaker nor long periods of laughter, can distinguish between an *interested* or *not interested* listener in a conversation.

The clearest distinction between the two diagrams are the nodes on the smallest pentagon (colour coded light blue) representing the listener's social response [V] (voiced). Voiced regions imply an exchange of short single words like 'uh-huh' or 'yea'. Voiced [V] is a vocal form of *backchannel response* [138, 85]. *Backchannel responses* are used by the listener to give feedback to the speaker, expressing acknowledgement, understanding, and presence in the conversation [108]. Here we see that a majority of these [V] nodes are bigger in the *interested* scenario when compared to the *not interested* scenario, especially in the rule [G-L] $\Rightarrow$ [V], when the speaker is gazing directly at the listener.

The next discernible trend belongs to the nodding nods [N] on the fifth pentagon from the centre (colour coded dark blue). Similar to voiced [V], nodding [N] is a visual form of *backchannel response*, used to confirm engagement in the conversation. Although the listener in the *not interested* scenario mirrors the speaker well in comparison to the *interested* scenario (i.e. [N] $\Rightarrow$ [N]), the listener in the *not interested* scenario barely nods in response to any other social signals. In this case, mirroring is not a discerning social behaviour between an *interested* and *not interested* listener. However, from the diagram we can see that the *interested* listener nods more consistently in response to the other social signals, especially when the speaker gazes directly at the listener (i.e. [G-L] $\Rightarrow$ [N]).

The final discernible trend is the talking social response [T] (second pentagon from the centre, colour coded red). While only a mild occurrence in the *interested* scenario, it rarely occurs at all in the *not interested* scenario. Talking [T] suggests *turn-taking* [51],

		Speaker							
		[L]	**[S]**	**[N]**	**[A]**	**[G-L]**	**[G-OL]**	**[G-N]**	**Aver**
Listener	**[V]**	**1.4**	**1.7**	**1.5**	**1.4**	**1.5**	**1.1**	**1.7**	**1.5**
	[T]	**2.2**	**3.9**	**14**	**2.2**	**2.7**	**2.5**	**4.4**	**4.6**
	[L]	0.9	1.2	1	1	0.8	0.9	1	1
	[S]	0.5	0.7	5	0.7	0.7	0.3	0.6	1.2
	[N]	**1.8**	**2**	**1.2**	**1.6**	**1.9**	**1.6**	**1.9**	**1.7**
	[A]	0.8	1.2	1.3	1.6	1.1	0.9	1.1	1.1
	[G-S]	0.9	1	1	1	0.9	1	1	1
	[G-OL]	0.8	0.9	0.7	0.7	1	0.7	1	0.9
	[G-N]	1.2	1.1	1.3	1.1	1.5	1	0.9	1.1

Table 5.4: **Ratio:** Ratio of *interested* to *not interested confidence* values for matching rules. An average is calculated in the last column.

whereby the listener attempts to participate in the conversation whilst the speaker is speaking.

From visualising the SDM, it is clear there are notable distinctions between an *interested* and *not interested* listener based on their exchanges in social signals with a speaker. To further validate these findings, using these *confidence* values, a ratio is taken of *interested* and *not interested* results for matching rules. This provides an indication of which listener's social responses play a leading role in discerning conversational interest. Results of greater than 1 is obtained when the rule occurs more frequently in the *interested* scenario, and less than 1 when they occur more in the *not interested* scenario. The results are shown in Table 5.4. Rows 1, 2, and 5 relating to the listener's social responses [V], [T], and [N] respectively, are the most prominent distinct trends between the two scenarios, with an average of greater than 1.5, as shown on the last column of Table 5.4. Also, rows 3 and 4 relating to the listener's social responses [L] and [G-S], produce an average of 1, varying equally in both the *interested* and *not interested* social scenarios. This is analogous to earlier observations.

The next step is to use these distinct social signals for conversation interest prediction.

5.4 Conversation Interest Predictions

Using only the discerning conversation social signals, the SDM is used to predict conversation interest. To perform this test, one person's social activity is eliminated from the data set. Data mining is then performed using only the other two people's social interaction with each other. This directly tests how well rules generalise between individuals. This is done separately for an *interested* and *not interested* scenario, resulting in two SDMs. These SDMs become the trained classifiers. SDM_{int} is the *interested* classifier, and SDM_{not} the *not interested* classifier. The eliminated person's social responses are then observed when in the role of a listener in both social scenarios. Using the SDM classifiers, we attempt to predict conversational interest based on the generalisation of rules across the subjects the model was trained on.

The prediction algorithm is as detailed in Algorithm 5.1. EL (Eliminated Listener) refers to the subject eliminated from the trained SDM classifiers, and when only in the role of listening to the other two subjects. TS (Trained Speaker) refers to the two other subjects the SDM classifiers were trained on, and when only in the role of speaking to EL. The predictions are done on the entire data set using different time frame windows ranging from 100 frames (4 seconds) to 7000 frames (approx $4\frac{1}{2}$ minutes) with 100 frame increments (denoted by $Incr$), resulting in 70 predictions overall. This highlights the optimal duration needed for predictions.

A single frame window $FrWin$, starts off as having a size of 100 frames, with a starting point (denoted as $StartPt$) at frame 1. All association rules within $FrWin$ and only between EL and TS are retrieved such that $\{TS_{FrWin}\} \Rightarrow EL_{FrWin}$. This results in a set of association rules $SetR$. The *confidence* values for matched rules between $SetR$ and the SDM classifiers, are summed separately for *interested* Sum_{int} and *not interested* Sum_{not} scenarios, such that $Sum = \sum \{conf(SetR_i \in SDM)\}_{i=1}^{|SetR|}$, where $conf(SetR_i \in SDM)$ denotes the *confidence* value for the rule $SetR_i$ that exists in SDM. A score of 1 is given to $Score_h$ (where h is the index for the current position of $FrWin$ within the footage), if Sum_{int} or Sum_{not} is larger for the scenario EL is in (within window $FrWin$), denoting a successfully prediction. Otherwise, a score of 0 is given denoting a failed prediction. However, if EL's interest level within window

Algorithm 5.1 Pseudo code for deriving conversation interest predictions

```
$MAX = 7000$, $Incr = 0$, $v = 0$
while $Incr \leq MAX$ {While query window size is less than $MAX$} do
  $Incr = Incr + 100$ {Increase query window size by 100}
  $StartPt = 0$, $h = 0$
  while $StartPt \leq (videolength - Incr)$ {While window remains within video} do
    $StartPt = StartPt + 1$ {Increment starting frame by 1}
    $FrWin = \{StartPt, ..., (Startpt + Incr)\}$ {Set query window position}
    $SetR \leftarrow \{TS_{FrWin}\} \Rightarrow EL_{FrWin}$ {Retrieve all association rules within window}
    $Sum_{int} = \sum \{conf_{int}(SetR_i \in SDM_{int})\}_{i=1}^{|SetR|}$ {Sum $conf$ for matching rules in $SDM_{int}$}
    $Sum_{not} = \sum \{conf_{not}(SetR_i \in SDM_{not})\}_{i=1}^{|SetR|}$ {Sum $conf$ for matching rules in $SDM_{not}$}
    if $EL$ is interested for all frames in window $FrWin$ then
      $h = h + 1$ {Increment $Score$ index by 1}
      if $Sum_{int} > Sum_{not}$ then
        $Score_h = 1$ {Give index $h$ a pass value of 1}
      else
        $Score_h = 0$ {Give index $h$ a fail value of 0}
      end if
    end if
    if $EL$ is not interested for all frames in window $FrWin$ then
      $h = h + 1$ {Increment $Score$ index by 1}
      if $Sum_{int} > Sum_{not}$ then
        $Score_h = 0$ {Give index $h$ a fail value of 0}
      else
        $Score_h = 1$ {Give index $h$ a pass value of 1}
      end if
    end if
  end while
  $v = v + 1$ {Increment $Prediction$ index by 1}
  $Prediction_v = \frac{|Score=1|}{|Score|} * 100$ {Compute overall precentage of passes}
end while
```

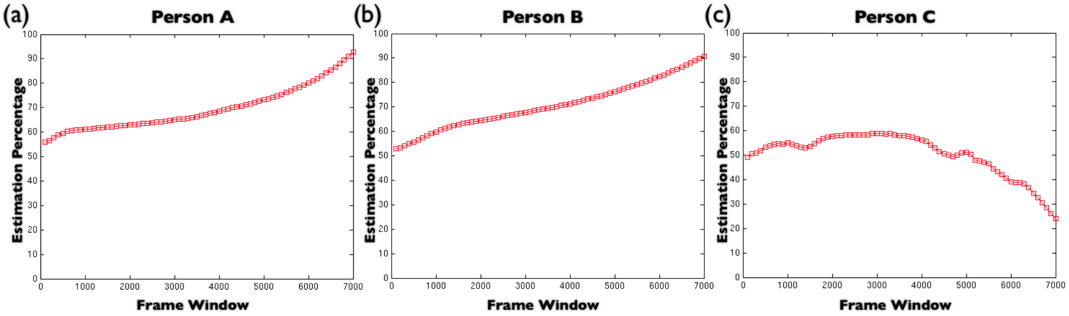


Figure 5.4: **Predictions:** Prediction percentage scores using varying frame windows for each person.

$FrWin$, is not constant in every frame (e.g. if the conversation changes within $FrWin$ and EL switches from *interested* to *not interested*), the window is ignored. This is repeated for $FrWin$ with different starting points $StartPt$, where $StartPt$ goes from 1 to $(videolength - Incr)$ with single frame steps. After $FrWin$ has reached the end of the footage, an overall percentage of successful predictions ($Prediction_v$) is computed, where v is the index for the respective window size used (determined by $Incr$), and $v = \{1, ..., 70\}$ relating to the 70 different window sizes. The entire process is then repeated using a different window size ($Incr = Incr + 100$), until a maximum window size (MAX) of 7000 frames is reached. There are three people in our data set, so we are able to perform this test three times (once for each person), alternating the eliminated listener. The results are shown in Figure 5.4.

As shown in Figure 5.4 (a), we are able to predict, with an accuracy of approximately 92%, person A's conversation interest, using a frame window size of 7000 frames. A similar result is achieved for person B, with a prediction accuracy of approximately 90%, also with frame window size of 7000 frames. However, with person C, the highest prediction of approximately 60% was achieved using a window size of 3000 frames (approx 2 minutes). Although the predicted accuracy of person C starts at a similar level to the other subjects for smaller time windows, the same characteristic rise in performance is not achieved as the window size increases.

Several conclusions can be drawn from this. Firstly, the indicative behaviour of the first two subjects could be said to be more consistent than between them and subject C. More subjects are required to investigate this. Another possibility is that mining separately on the derived *instances* of the *interested* and *not interested* scenarios, may result in the concealment or exaggeration of some *confidence* values. By mining these scenarios separately, the model makes the assumption that the *support* value $sup(R_i^A)$ (i.e. the probability of the occurrence of the set of speaker social signals), is similar in both scenarios. See Equation 5.2. A reasonable assumption is that a speaker's social behaviour is different when speaking to an *interested* and *not interested* listener. The aim of the experiment is to observe the *confidence* values (i.e. the conditional probability of a listener's social response given a set of speaker social signals) between the two scenarios. Therefore, $sup(R_i^A)$ should be constant for both.

By mining all *instances* together irrespective of the scenario, $sup(R_i^A)$ is the same in both, taking into consideration all the speaker's social exchanges, and better generalising over all their interactions. This increases the model's understanding of the speaker's general behaviour when considering *confidence* values. As such, a combination of antecedents that may have appeared less frequently in one or both *instances* when separated, may well appear more frequent (with respect to other antecedents) when combined, and visa versa. This will result in more accurate *confidence* representation, which takes into consideration the speaker's behaviour irrespective of the listener's social context.

In the next section, the alternative of computing the SDM using all *instances* together is investigated.

5.4.1 Predictions in a Combined Environment

To mine all *instances* together, new social signal labels are used, as shown in Table 5.5. The top row (Int List), are labels for the *person-independent interested listener*, the second row (N-Int List), are labels for the *person-independent not interested listener*, the third row are labels for the *person-independent speaker*, and the bottom row are their respective signals. By mining all *instances* together, $sup(R_i^A)$ will now take into consideration the entire conversation. A single set of *instances* are defined as $\digamma$ such

Int List	PI-1	PI-2	PI-3	PI-4	PI-5	PI-6	PI-7	PI-8	PI-9
N-Int List	PI-10	PI-11	PI-12	PI-13	PI-14	PI-15	PI-16	PI-17	PI-18
Speaker	-	-	PI-19	PI-20	PI-21	PI-22	PI-23	PI-24	PI-25
Signal	[V]	[T]	[L]	[S]	[N]	[A]	[GL]	[GR]	[GC]

Table 5.5: **Generalised labels for combined *instance*:** Set of generalised social labels to use for combined *instance*. The top row are labels for the *person-independent interested listener*, the second row are labels for the *person-independent not interested listener*, the third row are labels for the *person-independent speaker*, and the bottom row are their respective signals.

that $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^{N_I}$ where $\mathbf{f}_i$ is now a 25 dimensional binary vector.

$N_I = 67962$ *instances* were produced. 817 rules in total were extracted from the mining, 126 of which had single antecedents (i.e. $|R_i^A| = 1$), 323 with two antecedent, 321 with three antecedents, and 47 with four antecedents.

Figure 5.5 (b) shows the resulting new visual-SDM. In this new environment a different visualisation is used. Here, the nodes vary in diameter reflecting the ratio of mined *confidence* values between an *interested* and *not interested* listener for the given rule. For matching rules, the ratios are computed as the bigger *confidence* value relative to the smaller. For example, if the *confidence* value for PI-21 ⇒ PI-12 is greater than that for PI-21 ⇒ PI-3 (as defined in Table 5.5), then the ratio [N] ⇒ [L] is equal to $conf_{not}/conf_{int}$, where $conf_{not}$ is the *confidence* value for the *not interested* listener, and $conf_{int}$ the *confidence* value for the *interested* listener. When the node is blue, it means the confidence value for the given rule is greater when the listener is *interested.* Likewise, if the node is red, the confidence value for the given rule is greater when the listener is *not interested.*

The ratios are sorted by scale for simplicity. The relationship between the ratio-scales and node sizes are presented in Figure 5.5 (d). To make the visual comparison clearer, 1.0 is subtracted from all ratios so that, obtaining a value of 0.00 means the *confidence* values in both scenarios are equal, and the larger the value, the greater the difference in *confidence* between the two scenarios. Using this structure, we can more efficiently visualise prominent rules when comparing social scenarios.

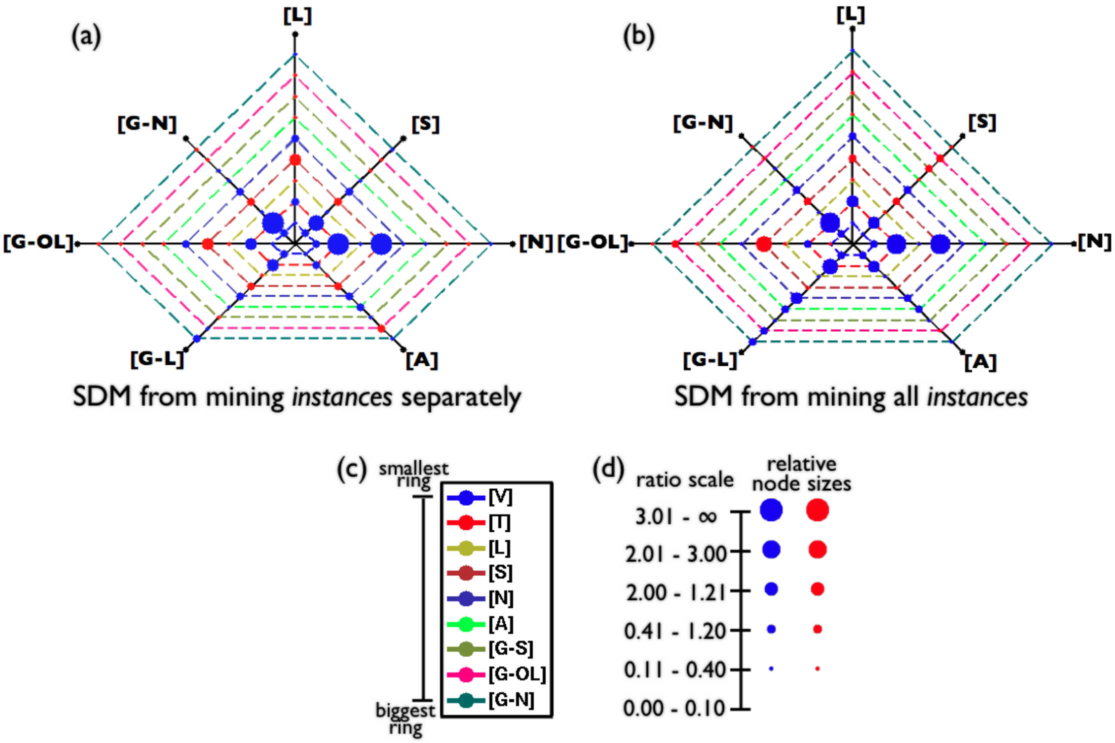


Figure 5.5: **New visualisation of SDMs:** (a) SDM from mining *instances* separately (b) SDM from mining all *instances*.

Figure 5.5 (a) shows this new visualisation for the SDM derived from mining the *instances* separately (as discussed in Section 5.3.1). Evidently, there are strong similarities between this SDM and the SDM derived from mining all *instances* together, as shown in Figure 5.5 (b). However, some nodes are slightly bigger in Figure 5.5 (b) in comparison to Figure 5.5 (a) (such as [G-OL] ⇒ [S]), and some are slightly smaller ([G-OL] ⇒ [T]). This suggests that, the *confidence* values for certain rules which were made smaller due to $sup(R_i^A)$ being computed separately for each social scenatio, are now bigger, and visa versa. Also, some nodes that were more prominent in one social scenario have now become more prominent in the other, such that the nodes have changed in colour (for example [S] ⇒ [G-N]). However, this is mostly the case for rules with very low ratios. It is important to mention that, although the differences between the SDMs in Figure 5.5 may appear minor when visualised, can still have a profound impact on predictions especially if the rule occurs regularly.

By visualising the ratios on a single plot, the distinctions between an *interested* and *not interested* listener become clearer. On both visual-SDMs in Figure 5.5, a majority of nodes, especially the larger ones, are blue, suggesting that *interested* listeners are overall more involved in the conversation than *not interested* listeners. However, there are more and larger blue nodes in Figure 5.5 (b) (mining on all *instances*). This is consistent with Pentland's [98] concept of using *texture* (i.e. energy) to predict positive and negative outcomes in social encounters. Although, the visual-SDM are also able to identify the distinct set of rules and multimodal social exchanges that influence such outcomes.

From observing the mirroring [97] nodes on both visual-SDMs (such as [N] ⇒ [N], [L] ⇒ [L], [A] ⇒ [A] etc), it is noticeable that they are all very small. This suggests that non-verbal imitation is approximately the same in both scenarios. Mirroring, also known as mimicry, is when people non-verbally and subconsciously imitate each other during dyadic interactions. Research in psychology has shown that regular occurrences of mirroring result in increased rapport and pro-social behaviour [72, 73]. Some also consider mimicry as a useful tool for predicting compatibility, connectivity and likeness between people [97]. One would expect mimicry to be a regular occurrence when the listener is *interested*, however, this is not the case. It is only fair to assume that mirroring is not an essential behaviour for discerning conversational interest.

The clearest distinction between the *interest* and *not interested* scenarios on both visual-SDMs, are the nodes on the second pentagon from the centre (colour coded red) representing the listener's social response [T] (talking). As explained earlier, talking [T] suggests turn-taking, which again has occurred more when the listener is *interested* in the conversation for all instances of the speaker's social signals. Unlike before (Figure 5.3), this distinction is much clearer due to this new visualisation using ratios.

In Figure 5.5 (b) (mining all *instances*), turn-taking is most prominent when the speaker nods [N], gazes at no one [G-N] and gazes at the listener [G-L], suggesting that these are the strongest visual cues for turn-taking. However, this is slightly different to Figure 5.5 (a) (mining *instances* separately), where turn-taking when the speaker head shakes [S] is considered more distinctive than when gazing at the other listener [G-L]. Previous psychological studies [10, 64], show that visual cues such as eye gaze and nodding, play an important role in conversational turn-taking, which agrees with the interpretation in Figure 5.5 (b).

When observing both visual-SDMs, with regards to this turn-taking pentagon (discussed above, second pentagon from the centre, colour coded red), it would appear that an *interested* listener is marginally more likely to turn-take when the speaker is gazing at no one [G-N] than when the speaker gazes directly at the listener [G-L]. Previous evaluations consider direct eye contact from the speaker [G-L] as the most essential gaze direction, for responsive and smooth turn-taking from the listener [127, 16]. Though this may be the case when observing the individual social scenarios, when discerning between the two, it is not. When a speaker is gazing at no one [G-N], in most cases this suggests *gaze aversion* [37, 80], which occurs when people are engaged in cognitive activity such as retrieving information from memory. This regularly involves a change in head orientation and gaze direction. As such, an *interested* listener is more astute in using this opportunity to turn-take, assisting the speaker and further participating in the conversation. Hence, when comparing both scenarios, the rule [G-N] ⇒ [T] is more acute than [G-L] ⇒ [T], and a more useful rule when discerning the interest of a listener.

From this new visualisation, it is also clearer that in the pervious approach of mining *instances* separately (Figure 5.5 (a)), the head shake [S] nodes on the fourth pentagon from the centre (colour coded brown), display highly discernible trends between an *interested* and *not interested* listener. On this pentagon, apart from [N] ⇒ [S], which shows a high trend in the *interested* scenario, all other nodes show a trend towards the *not interested* scenario. This is a similar characteristic on the visual-SDM from mining all *instances* (Figure 5.5 (b)), except the node [G-OL] ⇒ [S] (when the speaker gazes to the other listener, the listener shakes their head) is now considerably larger. Head shaking is traditionally known to convey contradiction, disapproval and disagreement. However, [N] ⇒ [S] occurs considerably more when the listener is *interested* in the conversation. Without the assistance of dialogue information, it is safe to assume when a listener responds to a nod [N] with a head shake [S], they are expressing a strong disagreement. For a listener to have an assertive opinion implies interest and insight

		Speaker							
		[L]	[S]	[N]	[A]	[G-L]	[G-OL]	[G-N]	**Aver**
Listener	[V]	**1.4**	**1.2**	**1.5**	**1.6**	**2.2**	**1.1**	**1.8**	**1.53**
	[T]	**2.2**	**2.9**	**14**	**2.6**	**3.9**	**1.9**	**4.6**	**4.54**
	[L]	1.1	1.2	1	1.1	1.1	1.4	1.1	1.15
	[S]	**2.1**	**2.1**	**6.5**	**1.3**	**1**	**3.3**	**1.6**	**2.56**
	[N]	**1.9**	**1.5**	**1.2**	**1.9**	**2.7**	**1.3**	**2**	**1.77**
	[A]	1.2	1.1	1.2	1.8	1.6	1.4	1.1	1.35
	[G-S]	1.1	1.4	1	1.1	1.3	1.2	1	1.2
	[G-OL]	1.2	1.5	1.4	1.3	1.6	1.7	1	1.38
	[G-N]	1.2	1.3	1.3	1.3	2.1	1.3	1.1	1.36

Table 5.6: **Scores for ratio using all *instances*:** Scores of taking the ratio of *interested* to *not interested* confidence values (as explained in Section 5.4.1) for matching rules. An average is calculated in the last column.

in the conversation, as opposed to a *not interested* listener who would be more passive and unresponsive. The other prominent rule is [G-OL] ⇒ [S], which occurs more when the listener is *not interested* in the conversation, produces a larger node when mined on all *instances*. [G-OL] means the speaker's gaze and attention is adverted from the respective listener. Hence, the speaker is not aware of what the respective listener is doing, which in this case is a head shake [S]. The performance of any gesture when the speaker is unaware of it, especially one commonly associated to disapproval, suggests a disconnection from the speaker and the conversation.

To better highlight the difference in mining using all *instances*, the average of ratios for the listener's social responses across all speaker's social signals are computed. This provides an indication of which listener's social responses now play a leading role in discerning conversational interest. The results are shown in Table 5.6. Similar to mining using separate *instances*, social responses [V], [T] and [N] are prominent distinct trends between the two scenarios. This is evident on the last column of Table 5.6, rows 1, 2, and 5, whereby they produce high averages of greater than 1.5. However, it is now clearer that [S] also provides distinct trends, even more prominent than the two

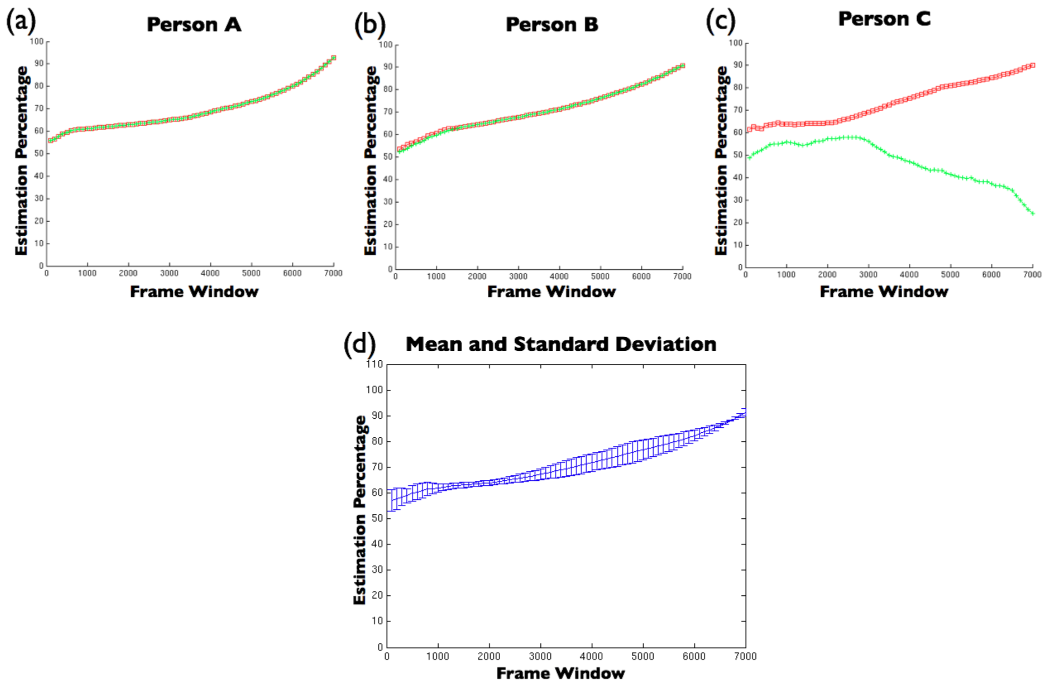


Figure 5.6: **Predictions using all *instances*:** In (a), (b), and (c), red lines represent prediction percentage scores using mining results of all *instances*, and the green lines are predictions from mining the *instances* separately and using the same 4 social responses. (d) overall average and standard deviation for results obtained from mining all *instances*.

backchannel responses [N] and [V].

As shown in Figure 5.6 (a), (b), and (c), using these highly discerning social signals obtained from mining all *instances*, person C's predictions have greatly improved following a similar characteristics to the other predictions. The new predictions are represented by the red lines and the previous predictions (using the same set of social responses) are represented by the green lines. Little improvements occurred for persons A and B.

Figure 5.6 (d) shows the overall average and standard deviation plot for the new predictions. With only 4 seconds of observation, predictions better than random are obtained. However, as more evidence accumulates, the performance increases to 90% for a time window of approximately $4\frac{1}{2}$ minutes. Theses results prove the SDM can derive distinct social trends between the two scenarios, which can generalise well for accurate predictions.

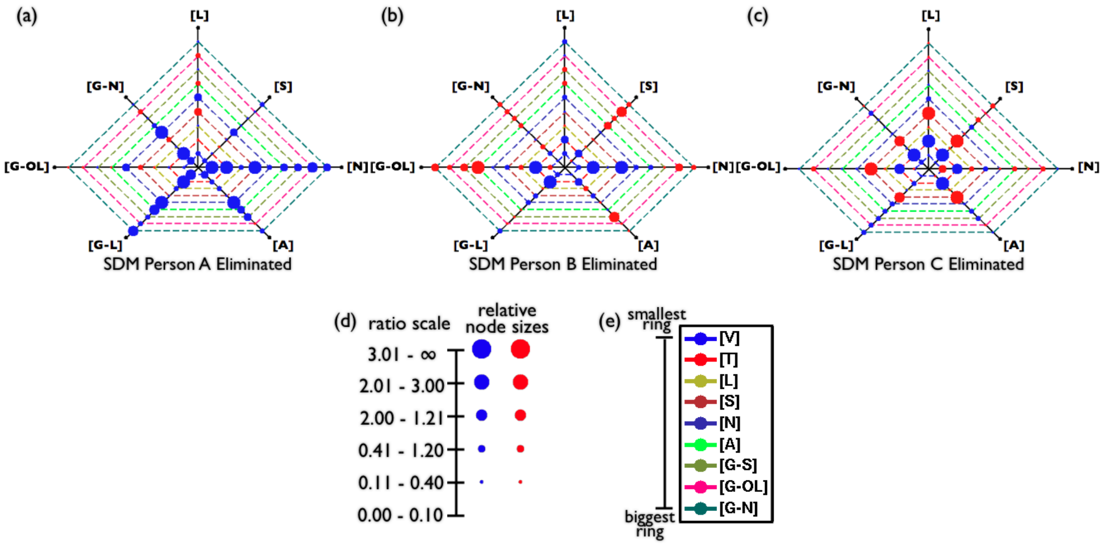


Figure 5.7: **Visual-SDMs from mining all *instances* used as classifiers:** (a) Visual-SDM with person A eliminated from the data set. (b) Visual-SDM with person B eliminated from the data set. (c) Visual-SDM with person C eliminated from the data set. (d) Relationship of ratio scale to node sizes. (e) Legend for pentagon rings.

5.4.2 Predictions with Varied Ratio Thresholds

Thus far, the visual-SDM has allowed social dynamics observation for determining prominent trends between an *interested* and *not interested* listener. With regards to the results obtained from mining all *instances*, 4 listener's social responses ([V], [T], [S] and [N]) where revealed to provide the most distinction between the two scenarios. These social responses were chosen by having ratio-averages greater than a ratio threshold η, where $\eta = 1.5$ (see last column of Table 5.6). Next, the outcome of varying threshold η on interest predictions from mining all *instances* are explored.

Figure 5.7 shows the 3 visual-SDMs used as the trained classifiers in Section 5.4.1, and Table 5.7 shows the ratio-averages for the SDMs. Even though the sample sizes are reduced from 3 to 2 subjects (with one person eliminated from the data set), the two strongest and most discerning social responses (turn-taking [T] and head shake [S]) still remain so. These numbers are highlighted in red. It is fair to assume that using only these two social responses will yield the most accurate predictions. However, there is

Listener	Ratio-Aver Elim A	Ratio-Aver Elim B	Ratio-Aver Elim C
[V]	2.49	1.43	1.48
[T]	**14.87**	**5.84**	**5.29**
[L]	1.56	1.45	1.24
[S]	**4.07**	**2.33**	**7.76**
[N]	3.41	1.47	1.35
[A]	1.93	2.10	1.55
[G-S]	1.60	1.96	1.41
[G-OL]	1.74	2.01	1.53
[G-N]	1.99	1.80	1.30

Table 5.7: **Ratio averges:** Table showing the ratio-averages for the 3 SDMs used as classifiers. The two strongest and most discerning social responses (turn-taking [T] and head shake [S], highlighted in red) from the observation of the entire data set still remain the strongest when observing the data set with smaller sample size (i.e. only 2 subjects).

no knowledge on how accurate these predictions are when all other social responses are involved. Starting with a threshold value of $\eta = 1.3$, η is gradually increased to observe the prediction averages when certain social responses are ignore that fall below the threshold. Results are shown in Figure 5.8.

Figure 5.8 (a), (b), and (c) shows the average predictions against varying ratio thresholds for each person. Figure 5.6 (d) shows the overall mean and standard deviation. Here, the ratio threshold is varied from 1.3 to 5. At the lowest ratio threshold, the lowest mean prediction of approximately 60% is obtained. As the threshold gradually increases, so does the mean prediction. The best mean prediction of approximately 80% is obtained at $\eta = 3.5$. At this threshold, only person A's [T] and [S] responses, person B's [T] response, and person C's [T] and [S] responses are used for predictions. Figure 5.8 (e), (f) and (g) detail the predictions at different time frame windows using $\eta = 3.5$. It is clear the prediction accuracies are higher overall when compared to Figure 5.6, especially for the smaller time frame windows. From this we can conclude that by increasing η, the prediction accuracy can be optimised. Next, the effects of varying the ratio threshold of specific rules (as opposed to varying ratio averages over a set of

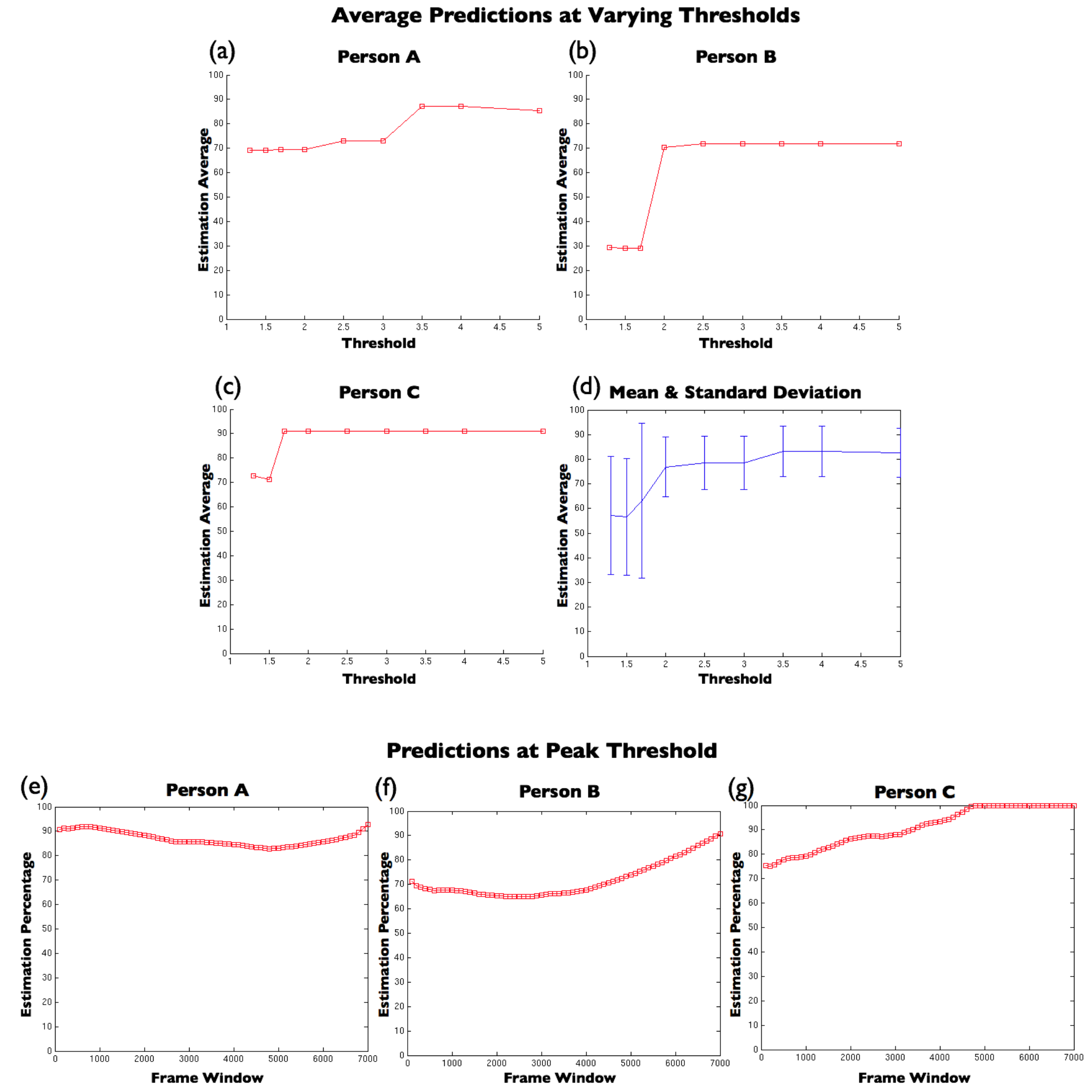


Figure 5.8: **Average predictions:** Average prediction percentage scores.

signals) are explored.

Figure 5.9 shows the results of varying ratio threshold η based on the ratios of specific rules. The blue line represents results when only considering single antecedents, and the red line considers all antecedents. The blue lines overall, show accurate predictions peaking at a threshold of 6. The single antecedent rules for the predictions of person C drops after a ratio threshold of 8. It can be assumed that at ratio 10, the respective rule(s) are unique to a specific scenario, but, are not regularly occurring or have high enough *confidences* to make an impact on the average predictions. At ratio 20, the prediction then drops to 0, since there are no rules with single antecedents that exhibited

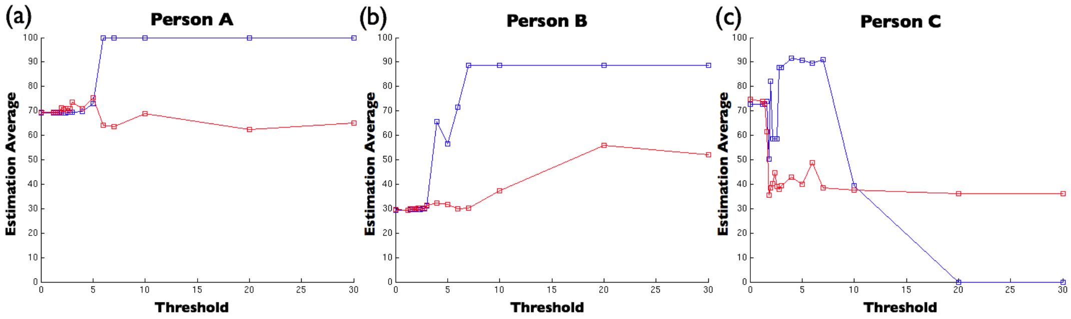


Figure 5.9: **Average predictions:** Average prediction percentage scores based on specific rules. Blue line for single antecedents and red line for all antecedents.

such high ratios.

When considering all antecedents (red lines), the predictions are generally not highly accurate. From this, the conclusion is drawn that the higher order antecedent (2 to 4) rules do not generalise as well as the single antecedent rules. This is most likely due to the different subjects personalities, since different people can perform a different combination of social signals when expressing the same emotion. For instance, as one subject may laugh, another may laugh and nod, and another may perform a more animated laugh with large body movements and hand gestures, and so forth. The single antecedent rules are not affected by this, since data mining treats them independent of the other antecedents. As such, using single antecedent rules generalises better because personality traits are not retained.

5.5 Summary

In this chapter, conversational experiments were conducted to identify the social dynamics between people in a conversation. Semi-supervised computer vision techniques are used to label social signals from the footage, and data mining employed to deduce frequently occurring association rules between the speaker and listener in both *interested* and *not interested* scenarios. Using the confidence values of rules, the SDM is built, which allows visualisation and prediction of conversational interests.

By performing mining in on all *instances*, the SDM can accurately predict conversation interest in less that 5 minutes of observation. Unlike knowledge driven methods [77, 107] that make prior assumptions on expected interactive behaviour, this approach is data driven, relying only on tangible rules governed solely by the data. This approach also extends on work on data driven models of social interaction [47], using multimodal mined association rules, making this approach simple and diversable to other social contexts.

An important point to mention is that, in this work, the questionnaires given to the subjects prior to data capture were used to deduce if they were *interested* or *not interested* in certain topics. These questionnaires were subsequently used as ground truth for the conversation interest predictions. However, there is a possibility that the subjects may have become interested in a topic even if they had no interest in it beforehand, and vica versa. Though this may result in some inaccuracies in the prediction, our results show that such discrepancies that contradict the ground truth is only a mild occurrence, and is alleviated with an increase in the observation period (up to $4\frac{1}{2}$ minutes window). More subjects are required to investigate this issue further, however, with prediction accuracies of over 90%, we can assume that this issue does not greatly impact the performance of the model.

In the next chapter, this approach is extended to provide autonomous control of socially interactive avatars. By combining the MIMiC system in Chapter 4 with this social model, the SDM drives the animations, whilst the user determines who speaks and the level of interest of the listeners. Such a social motion controller is intuitive, providing animators with the means of conducting character animations as opposed to dictating their articulated motion.

Chapter 6

Social Interactive Human Video Synthesis

In Chapter 4, the generative model for motion synthesis from Chapter 3 was extended to an example-based model, referred to as MIMiC, which allowed real-time, multimodal control of animations. As well as interactive animations, the model was demonstrated on a synthetic listener, which responded appropriately to a speaker's audio signal by nodding. This approach was intuitive but incorporated no contextual information. The synthetic listener would nod regardless of how it was spoken to, so long as there was an inherent voice inclination in the audio speech signal, which is always the case in human speaking patterns. To address this lack of contextual information, a social dynamics model (SDM) was devised in Chapter 5. The SDM used data mining to derive prominent association rules between a speaker and listener in both *interested* and *not interested* social scenarios. The *confidence* values from these rules could generalise well and predict conversational interest from a short period of video observation.

This chapter takes the next step, combining the MIMiC system with the SDM to generate socially interactive avatars. Additional work is required to make this integration possible. The MIMiC system is extended to provide more robust motion blending by incorporating a *texture motion graph*, specifically tailored to animate photorealistic human social behaviour. The SDM is also tailored to model person specific social behaviour as opposed to the generalisation previously proposed. This allows each avatar

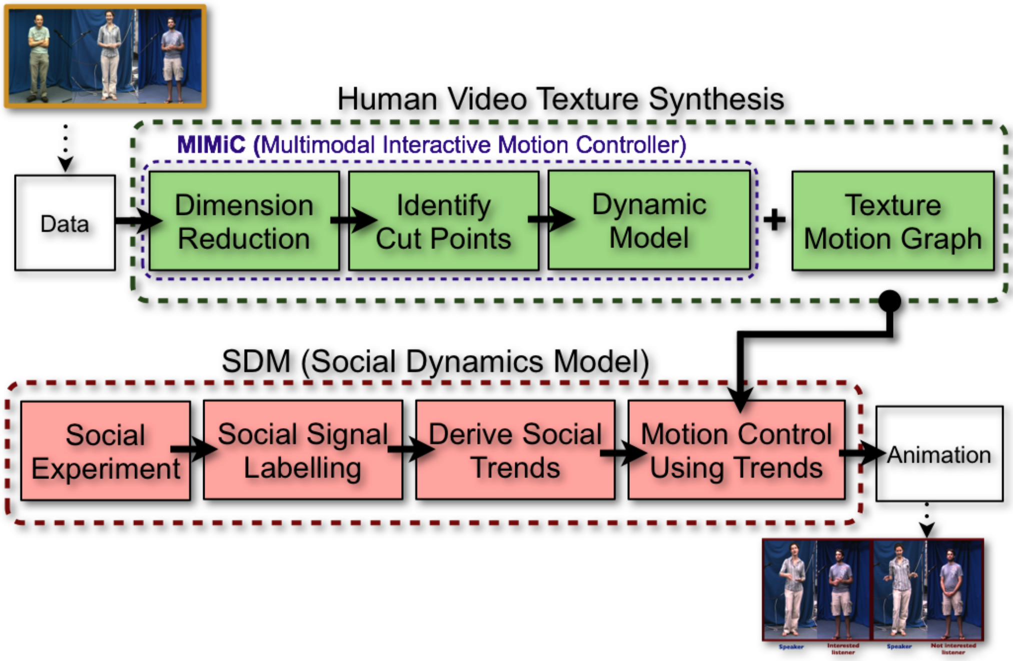


Figure 6.1: **System overview:** System overview for developing social interactive avatars. Consists of two main stages; *Human video texture synthesis* and *Social dynamics model*. The input data are the avatar video sequences. The texture synthesis stage consists of the MIMiC system (as presented in Chapter 4), and a texture motion graph which will be explained in Chapter 6.2.3. The social dynamics model is as detailed in Chapter 5, except person specific social labels are now used to retain the unique personalities of the avatars. The resulting output are the autonomous socially interactive avatars

to maintain their personal style of communication, making the resulting animations appear more natural.

This chapter is organised as follows: Section 6.1 presents an overview of the approach. Section 6.2 details the tailoring of MIMiC for animating human video textures. Section 6.3 explains the character specific SDM, and Section 6.4 combines both MIMiC and SDM to generate socially interactive avatars. The reminder of this chapter presents the results and conclusions are drawn.

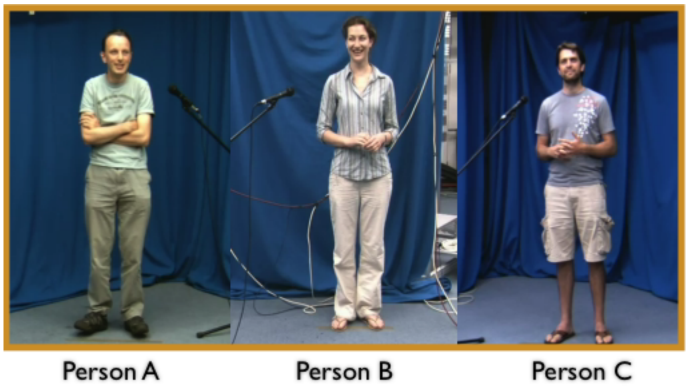


Figure 6.2: **Video data set:** Image showing full-body view of recorded video data of three people having a conversation.

6.1 Overview

Figure 6.1 presents the overview of this chapter, and how the framework is configured to generate socially interactive animations. Given a data set of full-body sequences of 3 subjects having a conversation, this data becomes the input to the MIMiC system. Combining MIMiC with a *texture motion graph*, the reliability of the generated transition points are increased, guaranteeing global connectivity of cut points to a majority of other unique motion types.

Using the SDM as discussed in Chapter 5, social trends are derived for person specific association rules. The *confidence* values extracted from these rules are used as conditional probabilities to drive MIMiC, resulting in autonomous socially interactive avatars.

6.2 Human Video Texture Synthesis

6.2.1 Data Set for Human Video Texture Synthesis

The data set is the same as that used in Chapter 5. It consists of approximately 30 minutes of video and audio recording of the full-body frontal view (516×340, 25 frames per second, 48kHz) of 3 individuals having a conversation with each other. This is shown in Figure 6.2. We refer to the 3 individuals as person A, B, and C.

Each full-body video consists of approximately 43000 frames. To reduce computational complexity, the videos were reduced to grayscale and resized to a quarter of their original size. Given a video sequence $\mathbf{X}$, each frame is represented as a vector $\mathbf{x}_i$ where $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{N_T}\}$ and N_T is the number of frames and $\mathbf{x}_i = (x_{i1}, y_{i1}, ..., x_{ix}, y_{iy}) \in \Re^{xy}$ for an x × y image.

To further reduce the complexity, Principal Component Analysis (PCA) is used for dimensionality reduction as described in Section 3.2. The dimension of the feature space $|x_i|$ is reduced by projecting into the eigenspace d, where d is the chosen lower dimension $d \leq |x_i|$ such that $\sum_{i=1}^{d} \frac{\lambda_i}{\Sigma \forall \lambda} \geq .98$ or 98% of the energy is retained. $\mathbf{Y}$ is defined as a set of all points in the dimensionally reduced data where $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_{N_T}\}$ and $\mathbf{y}_i \in \Re^d$.

Person A was reduced to $\Re^{476}$, person B to $\Re^{399}$, and person C to $\Re^{241}$.

6.2.2 Identifying Candidate Cut Points

The unsupervised motion segmentation approach in Section 4.3.1 is used to identify cut points and segment the motion data into shorter subsequences. Using this approach is highly effective especially when dealing with large data sets, adopting a k-medoid cluster algorithm to define N_C k-medoid points as suggested in Section 4.3.1. By only computing the L2 distance at these points, we reduce the amount of computation required to define candidate transitions, focusing attention on regions where transitions are most likely.

Here, a distance matrix is computed for each subject to give an idea of how different they are with regard to their body language. This is shown in Figure 6.3. It is evident that person B produces the most varying postures, most likely due to frequent use of highly exaggerated hand gestures and body movements during social interaction. Although finding appropriate transition points for this subject may be more challenging, the non-verbal social signals performed by their resulting avatar will be highly energetic,

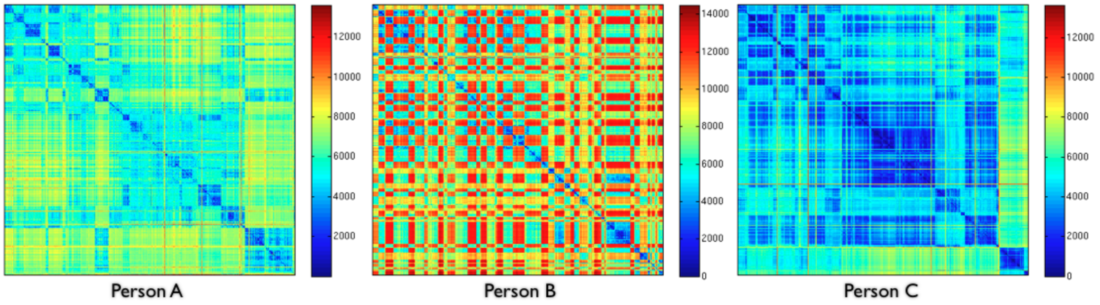


Figure 6.3: **Distance matrix:** Distance matrix of texture data set for each subject.

noticeable, and more convincing. Person C however keeps their movements and gestures to a minimum. This is apparent by the large amount of similar frames represented by the blue regions. Deriving similarity for this avatar will be relatively easy, though their non-verbal social signals may appear very quick, subtle and difficult to notice. Some avatars may be more responsive or aesthetically pleasing than others, based on the availability of appropriate transition points and their own style of communication.

To preserve dynamics and account for temporal shape similarity, a linearly weighted average of similarity over a fixed window of 0.25 seconds is computed, centred on the k-medoids. Using a user defined threshold θ, the nearest points to each k-medoid points are identified to form clusters of cut points. The set containing the cut points of the n^{th} cluster is defined as $\mathbf{Y}_n^c = \{\mathbf{y}_{n,1}^c, ..., \mathbf{y}_{n,Q_n}^c\}$, where the number of cut points of the n^{th} cluster is denoted as Q_n. As explained in Chapter 4, Section 4.3.1, N_C is empirically determined based on the number of candidate cut points versus the quality of transitions and the amount of computation. In this work $N_C = 165$, reducing the number L2 distance calculations.

Formally, each cut point belongs to a single cluster where plausible transitions can be made between group members. This approach works well in data sets with high connectivity, however, less so for human video data where connectivity is limited. To overcome this problem, the approach is extended to allow cut points to belong to more than one cluster, providing the cut point clusters with more opportunities to perform novel movement.

Here, the transitions from the n^{th} cluster are of the form $\{(\mathbf{y}_{n,1}^c, \mathbf{z}_{n,1}^c, \iota_{n,1}, C_{n,1}^z), ...,$

$(\mathbf{y}^c_{n,Q_n}, \mathbf{z}^c_{n,Q_n}, \iota_{n,Q_n}, C^z_{n,Q_n})\}$, where $\mathbf{y}^c_n$ is a cut point in the n^{th} cluster acting as the start transition point, $\mathbf{z}^c_n$ is the end transition point denoting the end of the subsequence between $\mathbf{y}^c_n$ and $\mathbf{z}^c_n$ (where $\mathbf{z}^c_n \in \mathbf{Y}^c$ and $\mathbf{z}^c_n \neq \mathbf{y}^c_n$), ι_n is the frame number of $\mathbf{y}^c_n$ in the original data, and C^z_n is the index of the cluster $\mathbf{z}^c_n$ belongs to.

6.2.3 Texture Motion Graph

The cut points can be used to transition between different subsequences available in the data. However, there is no consideration to whether cut points can access all the available motion types. Also, although dead-ends can be avoided by the destination driven dynamic programming, as discussed in Chapter 4, Section 4.3.4, their presence still presents a risk. As a result, a *texture motion graph* is pre-computed to guarantee global connectivity to different types of movements in the video data set.

Motion graph, proposed by Kovar et al. [70], essentially connects various subsequences together to a form a directed graph, whereby the edges are the generated cut points. By assembling the graph, we can identify and eliminate cut points and cut point groups with low connectivity, improving reliability in the sequence selection process. Various forms of motion graph have been proposed in recent years [13, 111, 101, 17], built for animating motion captured data. *Texture motion graph* is specifically tailored for video textured data and extended to overcome ambiguities in transitioning between video frames.

Generating smooth blends between human video textures is a very challenging topic, since as human beings, we can easily recognise unnatural human movement or textures. Similarity measure performs well at distinguishing different body poses, however does not account for facial gestures like laughing, talking and subtle changes in gaze direction. To overcome this, the social signal annotations derived in Chapter 5, Section 5.1.3, are used to assist the similarity measure. Figure 6.4 demonstrates the approach. Given a distribution of candidate cut points and cut point groups represented by the green dots and the circle encompassing them respectively in Figure 6.4 (A). Cut points that do not have the same gaze, talking, and laughing labels, as those of its k-medoid cluster centre, are represented by the red points in Figure 6.4 (B), and are pruned from

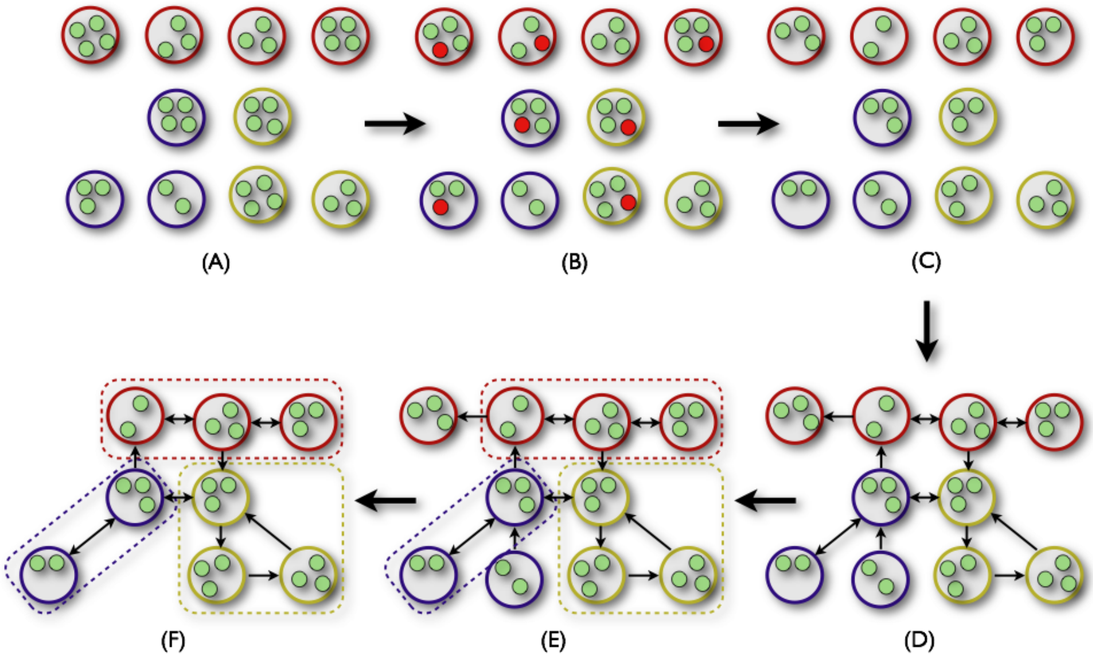


Figure 6.4: **Example of texture motion graph process.** (A) Network of cut points and cut point groups. (B) The cut points highlighted in red do not have the same gaze, talking, and laughing labels, as those of its k-medoid cluster centre. (C) The red cut points are pruned from the network. Removing them reduces the occurrence of rapid and unatural changes in facial expressions during transitions. (D) Cut point groups of similar motion types are coded by colour. The arrows represent the directions of possible transitions between cut point groups. (E) Cut point groups of similar motion types are segmented into strongly connected subgraphs. (F) Groups that are not within a strongly connected subgraph are removed, mitigating dead-ends and guaranteeing more robust transitions.

the network in Figure 6.4 (C). Removing these cut points reduces the occurrences of rapid and unnatural changes in facial expressions and gaze direction during transitions.

A ‘strongly connected’ subgraph denotes a direct graph whereby a path exists to and from every cut point group. Traditionally, a single strongly connected subgraph is created for each type of motion, where each subgraph is either automatically linked or manually linked using post process linear interpolation [70, 13]. This approach is better suited to motion captured data since linear blending can disguise ambiguities in transitions between manually linked subgraphs. In human video data, links between subgraphs are best done by traversing through the original data, relying as little as possible on linear blending which can cause blurring. Hence, an increased level of connectivity is needed.

As opposed to just a single graph, n strongly connected subgraphs are generated for each unique set of social signals. This is demonstrated in Figure 6.4 (D), where the colour coded cut point groups represent similar social signals, and the arrows indicate the possible directions of transitions between cut point groups. The dashed lines encompassing the groups in Figure 6.4 (E) represents strongly connected subgraphs. As shown in Figure 6.4 (F), by pruning cut point groups that do not fall within a strongly connected subgraph, this structure mitigates potential dead-ends. These subgraphs can then connect to other subgraphs resulting in a strongly connected overall graph.

This structure efficiently populates the graph with various links to social behaviour, making them easily accessible from any subgraph pose configuration. Social signals, such as head nods and head shakes, can occur in short quick bursts, lasting only a few seconds. By making available varying occurrences of a set of labels, we increase the opportunity of transitioning to a social behaviour quickly, and easily, making the graph more responsive. The Tarjan algorithm [121] is used to derive the strongly connected subgraphs, and in our experiments we found $n = 3$ sufficient to populate our graph with varying social behaviour.

6.3 Character Specific SDM

In Chapter 5, the SDM was used to generalise human social dynamics for the purpose of conversational interest predictions. To demonstrate the diversity of the model, in this section, the SDM is extended to drive animations of social interactive avatars.

To allow each avatar to retain the personality of their respective subjects, instead of mining for generalised social signals, *person specific* social labels are used. This is shown in Table 6.1. The mining is performed separately for both *interested* and *not interested* scenarios. As before, both voiced [V] and talking [T] signals for speakers are ignored. Here, two sets of *instances* for *interested* and *not interested* scenarios are defined as

Per. A	**List**	U-1	U-2	U-3	U-4	U-5	U-6	U-7	U-8	U-9
	Spk	-	-	U-10	U-11	U-12	U-13	U-14	U-15	U-16
Per. B	**List**	U-17	U-18	U-19	U-20	U-21	U-22	U-23	U-24	U-25
	Spk	-	-	U-26	U-27	U-28	U-29	U-30	U-31	U-32
Per. C	**List**	U-33	U-34	U-35	U-36	U-37	U-38	U-39	U-40	U-41
	Spk	-	-	U-42	U-43	U-44	U-45	U-46	U-47	U-48
Signal		[V]	[T]	[L]	[S]	[N]	[A]	[G-S]	[G-OL]	[G-N]

Table 6.1: **Set of *person specific* social labels:** Persons A, B and C can each take the role of listener and speaker in the conversation. The mining is done separately for *interested* and *not interested* scenarios, with the person who is speaking and listening known.

$\mathcal{F}_{(int)}$ and $\mathcal{F}_{(not)}$, such that $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^{N_I}$. $\mathbf{f}_i$ is a 48 dimensional binary vector, where $1-16$ are for person A, $17-32$ for person B, and $33-48$ for person C. As explained earlier, a single *instance* is the interaction between 1 speaker and 1 listener. However, using these labels, the person speaking and listening are known.

The objective in Chapter 5 was to generalise over the SDM to perform conversation interest predictions of a listener. Mining was performed on all person independent *instances*, treating the generic speaker's social signals as independent of the listener's interest level. The aim in this chapter is to model the unique relationships between the individuals in roles of both speaking and listening. As such, in order to retain specific dynamic information, mining is performed separately. By mining separately, an assumption is made that the speaker's social signals are influenced by the listener's conversation interest level. Although very subtle and difficult to measure, this leads to the expectation that the listener's social responses are influenced by the speaker's knowledge of their interest in the conversation. For example, if person A can tell that person B is *not interested* in the conversation, they may well gaze more at person C. Person B's social responses are therefore as a result of person A's mild neglect. These more subtle relationships are very unique to the individuals, and mining the *instances* together irrespective of the scenario, would generalise over these specific exchanges.

Apriori Association mining is applied to the labels for both *interested* and *not interested* scenarios separately. The set of all rules extracted using data mining are defined as:

$$R = \{(\{R_i^A\} \Rightarrow R_i^C, conf_i)\}_{i=1}^{|R|} \tag{6.1}$$

where the total number of rules is $|R|$. As proposed in Chapter 5, *temporal bagging* ($s = 10$) is used to enforce temporal coherence between features.

6.4 Social Interactive Animation using Apriori Mining

The idea is to animate both the speaker and listener, but the SDM only drives the animations of the listener. As such, the conditional probability of the listener's social response given the speaker's social signals is computed as weighted variables to control the human texture synthesis model.

Given the chosen speaker's *person specific* 7 dimensional binary vector $\mathbf{f}_t$ (as explained in Section 6.3) at time t, where $\mathbf{f}_t \subset \mathbf{f}_\iota$ (ι is the frame index of the current query cut point as detailed in Section 6.2.2), the power set $2^{\mathbf{f}_t}$ for all combinations of the speaker's active social signals are derived. We find a suitable matching set of rules $R^t \subset R$ such that $\forall(\{R_j^{A,t}\} \Rightarrow R_j^{C,t}, conf_j) \in R^t$, where there exists $\mathbf{f} \in 2^t$ and $\mathbf{f} \subset R_j^A$. The weighted combination of the results are obtained as follows:

$$W = \sum_{j=1}^{|R^t|} conf_j I(R_j^{C,t}, \mathbf{f}_\iota) \tag{6.2}$$

where

$$I(R_j^C, \mathbf{f}) = \begin{cases} 1 & \text{if } \mathbf{f} \subset R_j^C \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

In the motion synthesis process as proposed in Chapter 4, Section 4.3.3, Equations 4.1 is altered as follows:

$$\boldsymbol{\phi}_m = P(C^z_{C_{t+1},m}|C_t).P'(\mathbf{z}^c_{C_t,m}).W \tag{6.4}$$

where $P(C^z_{C_{t+1},m}|C_t)$ is the probability of transitioning from the current cut point group C_t to the query cut point group C_{t+1}, m, and $P'(\mathbf{z}^c_{C_{t+1},m})$ is the likelihood of the respective end transition point.

6.5 Animation/Results

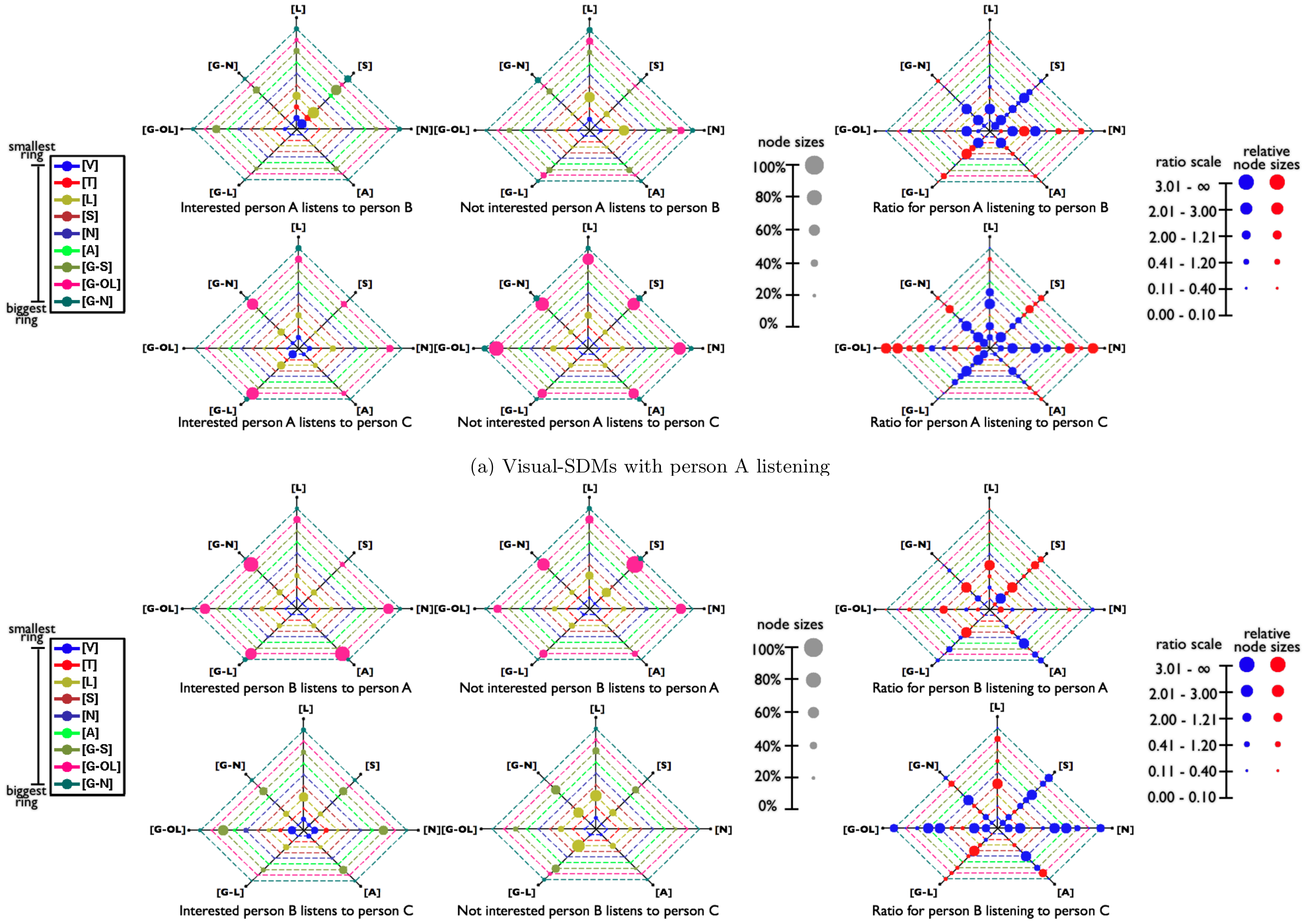
6.5.1 Person Specific Visual-SDMs

As explained earlier, the mining is performed separately using the *person specific* social signal labels as detailed in Table 6.1. 27127 *instances* were generated for the *interested* scenario, whereby data mining extracted 1323 rules in total. 366 had single antecedents, 706 with two antecedents, 240 with three antecedents, and 11 with four antecedents. 26230 *instances* were generated for the not interested scenario. In this scenario, data mining extracted 1373 rules, where 359 had single antecedents, 703 had two antecedents, 299 had three antecedents, and 12 had four antecedents. This resulted in 1034 matching rules in both scenarios, whereby, rules that were not prominent in both scenarios are ignored.

Figure 6.5 and 6.6 present the resulting visual-SDMs. Here, the visual-SDMs are displaying the social dynamics between 1 specific speaker and 1 specific listener, as opposed to a generalisation presented in Chapter 5. The visual-SDMs are presented in both the discrete form (as explained in Chapter 5, Section 5.2) and the combined ratio form (as explained in Chapter 5, Section 5.4.1). The discrete forms are on the first two columns of Figure 6.5 and 6.6, allow visualisation of the relative sizes of the *confidence* values in both scenarios. The combined ratio form on the last column on the right, allows visualisation of social responses which are most distinctive between the two scenarios.

Figure 6.5 (a) shows the visual-SDMs for person A listening to person B (top row) and person C (bottom row). Figure 6.5 (b) shows for person B listening to person A (top row) and person C (bottom row). Figure 6.6 (a) shows for person C listening to person A (top row) and person B (bottom row). By observing the combined ratio visual-SDMs on the third columns of Figure 6.5 and 6.6, it is clear that in a majority of cases, there are more large blues nodes than red, suggesting that an *interested* listener is more active in the conversation than a *not interested* listener. This is especially the case for the listener's social responses [V] and [T], which is consistent with the findings of the generalised social trends in Chapter 5.

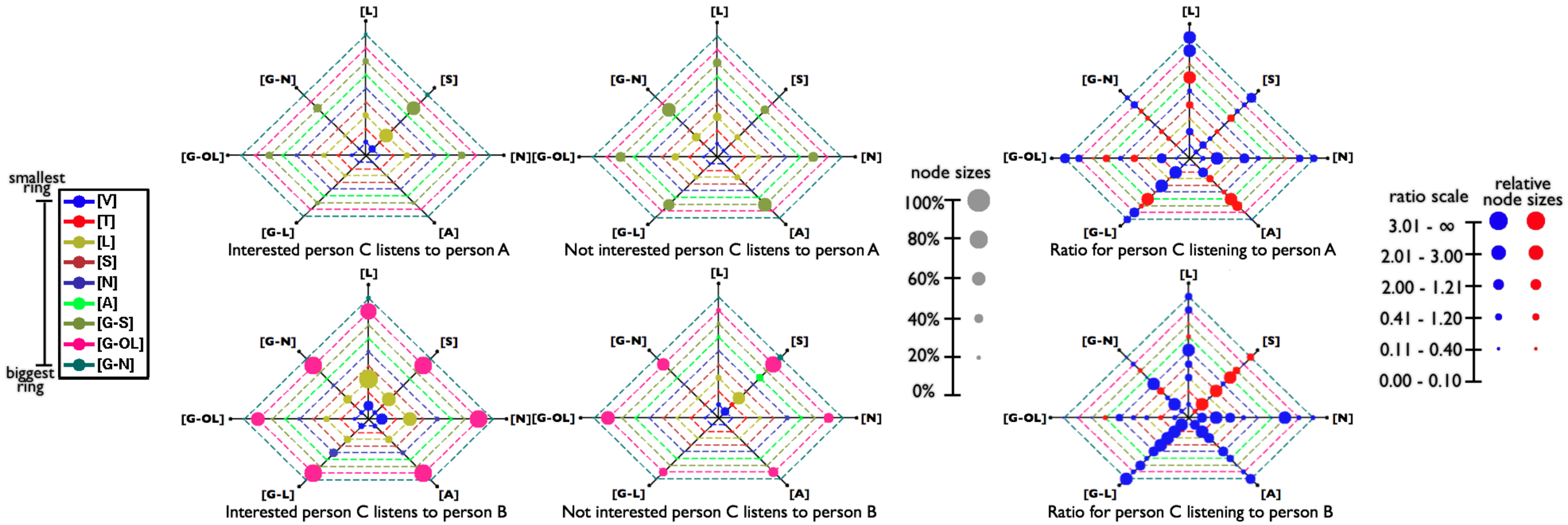
With regards to the ratio Visual-SDMs (third column on Figure 6.5 and 6.6), in general,



(a) Visual-SDMs with person A listening

(b) Visual-SDMs with person B listening

Figure 6.5: **Person specific visual-SDMs:** (a) Person A listening to person B (top row) and person C (bottom row). (b) Person B listening to person A (top row) and person B (bottom row). The visual-SDMs are presented in both the discrete form (first 2 columns) and the combined ratio form (last column).



(a) Visual-SDMs with person C listening

Figure 6.6: **Person specific visual-SDMs:** (a) Person C listening to person A (top row) and person B (bottom row). The visual-SDMs are presented in both the discrete form (first 2 columns) and the combined ratio form (last column).

the social response [N] is more prominent in the *interested* scenario, and the social response [S] (with the exception of [N]⇒[S]) is more prominent in the *not interested* scenario. Also, the nodes on the outer pentagons, which relate to gaze direction, tend to be smaller in comparison to the nodes nearer the centre of the pentagon. These are also consistent with earlier finds of the generalised case (see Figure 5.5).

The clear exception to these trends would suggest the unique behaviour of the specific subjects. These are presented for each person as follows:

- **Person A Listening:** As detailed on Figure 6.5 (a), with the exception of two cases ([G-L] ⇒[N] and [A] ⇒[S] when listening to person B), person A head shakes [S] more in the *interested* scenario when listening to both person B and C, for all occurrences of the speaker's social signals. It can be assumed that [S] is frequently used by person A when positively engaged in the conversation.

  Also, when person A is listening specifically to person C, their gaze behaviour is highly distinctive. When person C gazes at person B [G-OL], person A, when *not interested*, would more frequently also gaze at person B [G-OL] or gaze at no one [G-N]. Likewise, when person C nods [N], person A more frequently either gazes at no one [G-N] or gazes back at person C [G-S] when *not interested* in the conversation. Such prominent and specific characteristics between person A and C would require more study to evaluate. However, it can be assumed that when person A is listening to person C, gaze direction is an important social signal in determining interest.

- **Person B Listening:** When person B is listening to person A (Figure 6.5 (b)), person B tends to be more voiced [V] when *not interested* in the conversation. Though the nodes are not large suggesting low distinction, this can be considered as an act of encouragement from person B for person C to continue speaking. This might have been necessary if person B had been expressing dis-interest in the conversation which person C had noticed, As such, person B is putting more emphasis on expressing engagement. Again, this is difficult to interpret without additional experimentation.

- **Person C Listening:** Person C tends to respond very distinctively to when person B head shakes [S] and gazes directly at them [G-L] (see Figure 6.6 (a)). Whilst person B is speaking and head shakes [S], person C tends to turn-take [T], mirror/mimc [S], and perform more gestures [A] when *not interested*. Additionally, when person B is gazing at person C, person C performs a majority of the social responses more when *interested* in the conversation. This behaviour is also unique.

There are other unique characteristics between the subjects, but only the clearest have been discussed above. However, more study and experimentation is needed to evaluate these unique behaviours and draw justifiable conclusions.

In Chapter 5, trends in social dynamics between the subjects prove to generalise well, showing that both the visual and vocal *backchannel responses* [N] and [V], turntaking [T], and head shakes [S], produce the most distinctive trends. However, mining done on person specific social labels retains the personality of each subject and the unique social dynamics when conversing with each other. It is expected that in the majority of cases, these previously derived generalised trends will exist in the unique relationships between the subjects, but not for all. The objective here is to retain each subject's personal style of communication. By observing the unique social dynamics between the subjects using the visual-SDM, the resulting animations of the social interactive avatars can be validated.

6.5.2 Human Video Avatars

Using MIMiC, the user is given control over the various combinations of social behaviour of the human video avatars, however, not all combinations of social behaviour are possible. This is the case for all avatars with regards to performing head shakes [S]. This is mostly due to the limited availability of the particular social behaviour in the data set, resulting in very limited connectivity in the texture motion graph. Regardless, most of the popular combinations of social behaviour like *laughing* and *head nods* are connected and responsive.

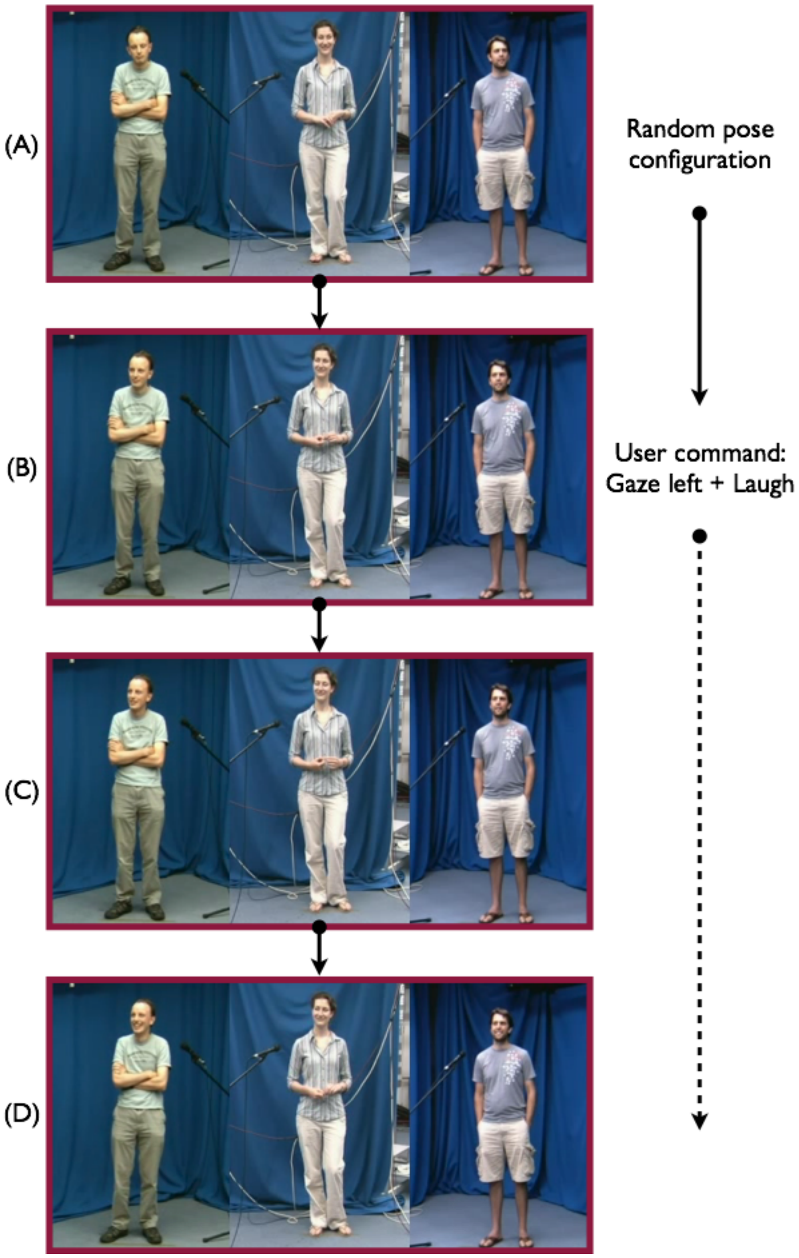


Figure 6.7: **User direct control of avatars:** From a random pose configuration, user gives each avatar a command to gaze left and laugh. MIMiC animates the best set of subsequences to perform the user specified activity.

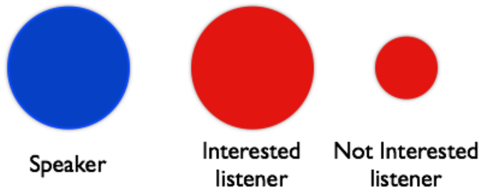


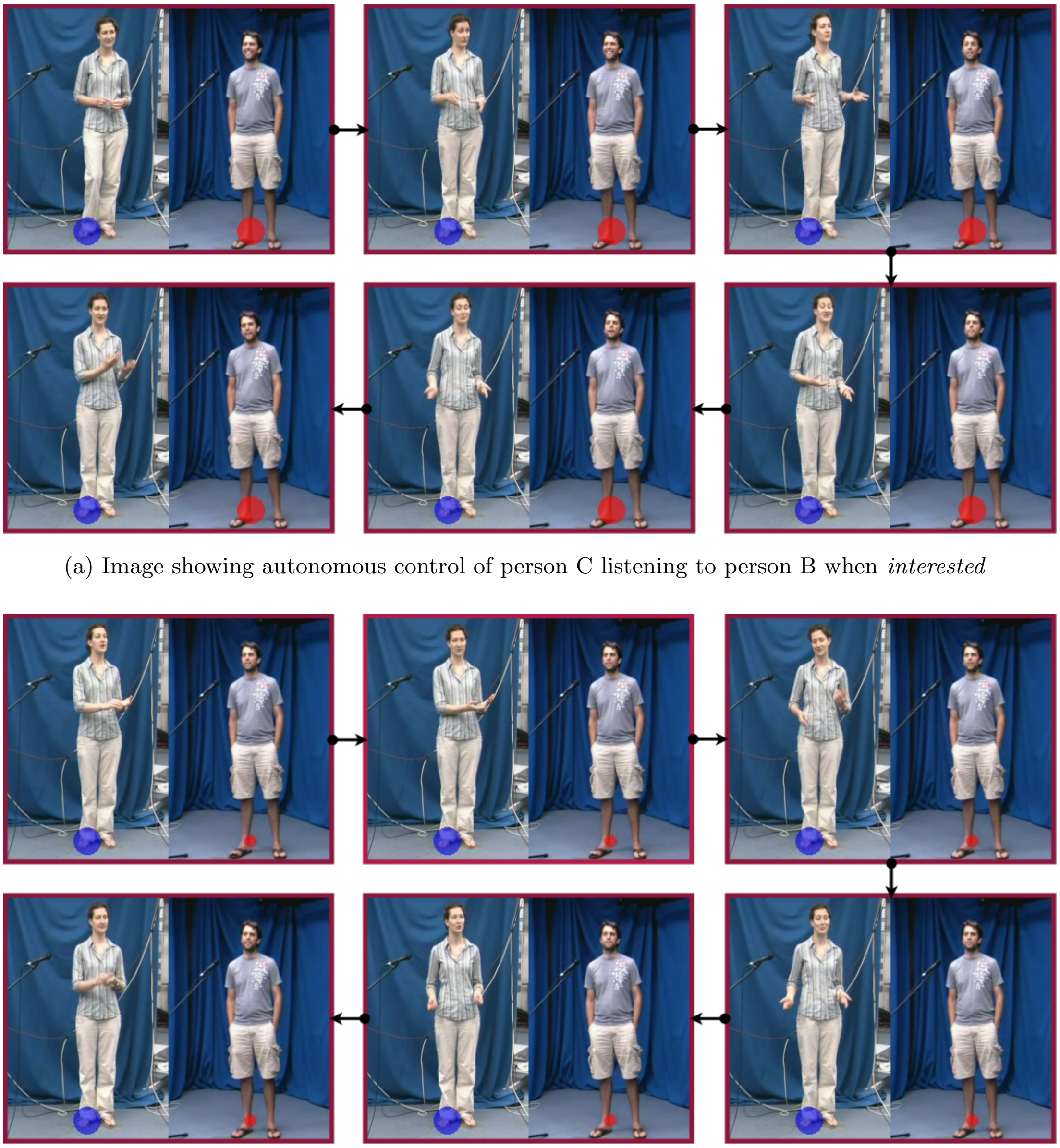
Figure 6.8: **Conversation Dynamic Key:** For clarity, a *conversation dynamics key* is used to indicate who is speaking, listening, and the listener's level of interest.

To demonstrate user control, as shown in Figure 6.7, from a random pose configuration (Figure 6.7 (A)), each avatar was instructed by the user to gaze left and laugh (Figure 6.7 (B) to (D)). MIMiC animates the best set of subsequences to perform the user specified activity. Each avatar is modelled independently using MIMiC, as a result, they respond to user commands at slightly different times. This is dependent on the current length of the subsequence the avatar is undergoing before the given command, as well as the accessibility in transitioning to the user specified action.

Not only can the user control the social behaviour of the video avatars but, using the data mined *confidence* values, a listener's social response in a given scenario can be automatically generated given the speakers social signal. For clarity, a *conversation dynamic key*, as shown in Figure 6.8, is used to indicate who is speaking, listening, and the listener's level of interest.

Both the speaker and listener are modelled using MIMiC, although, only the listener is generated autonomously using the mined *confidence* values. The speaker is given a single command [gaze at listener + speak], and the best set of subsequence are generated to perform the action. However, the listener's social responses are generated as a conditional probability given the speaker's social signals. The listener's pose configuration starts from random. The user can then select if the listener is *interest* or *not interested*, and the SDM drives the motion model to generate the most suitable set of subsequences in response to the speaker. This is demonstrated in three examples as shown in Figures 6.9, 6.10, and 6.11.

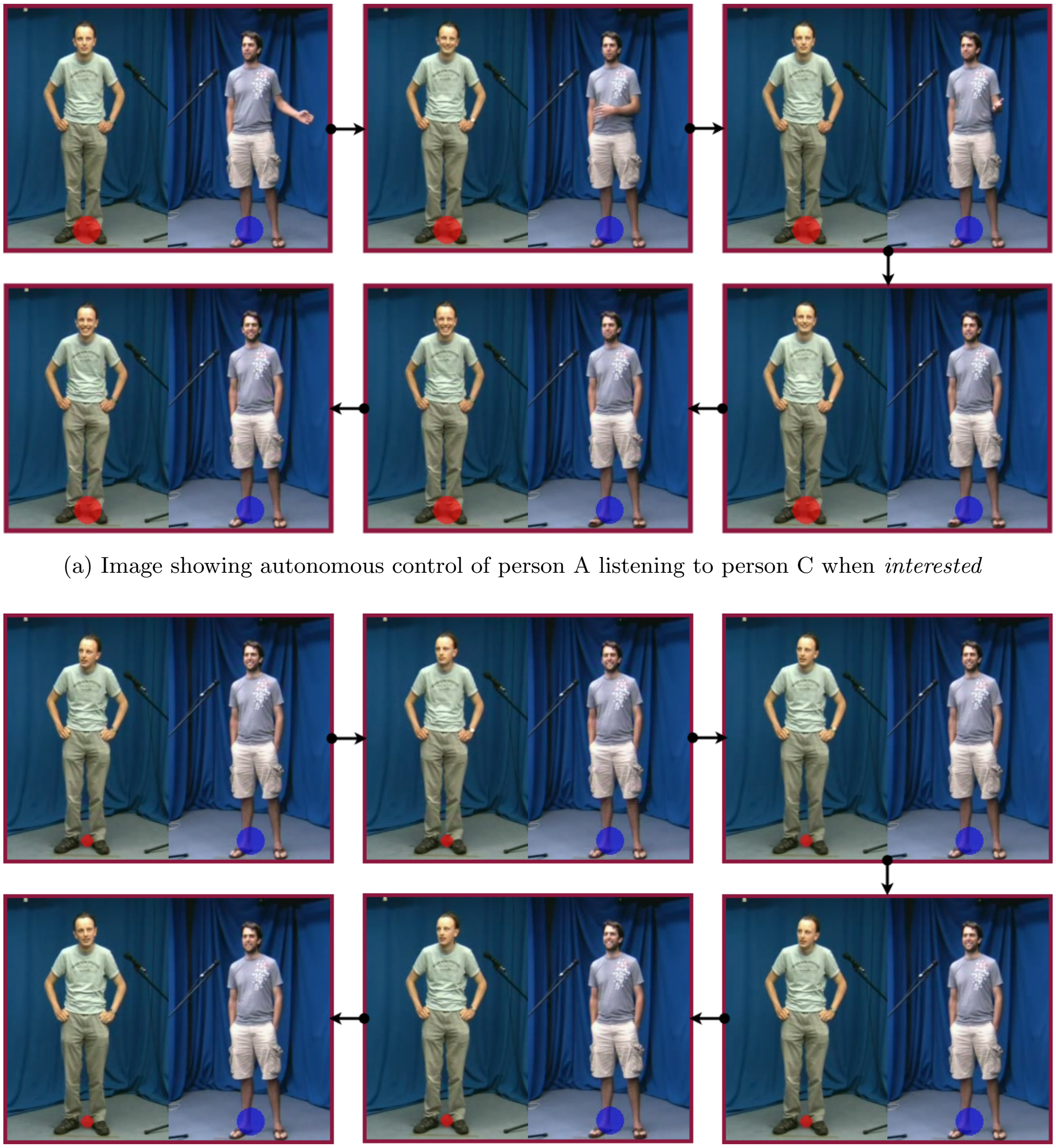
Figure 6.9, demonstrates autonomous control of person C listening to person B, when *interested* (Figure 6.9 (a)) and *not interested* (Figure 6.9 (b)) in the conversation.



(a) Image showing autonomous control of person C listening to person B when *interested*

(b) Image showing autonomous control of person C listening to person B when *not interested*

Figure 6.9: **Autonomous control example 1:** (a) Autonomous control of person C listening to person B when *interested* in the conversation (b) Autonomous control of person C listening to person B when *not interested* in the conversation.



(a) Image showing autonomous control of person A listening to person C when *interested*

(b) Image showing autonomous control of person A listening to person C when *not interested*

Figure 6.10: **Autonomous control example 2:** (a) Autonomous control of person A listening to person C when *interested* in the conversation (b) Autonomous control of person A listening to person C when *not interested* in the conversation.

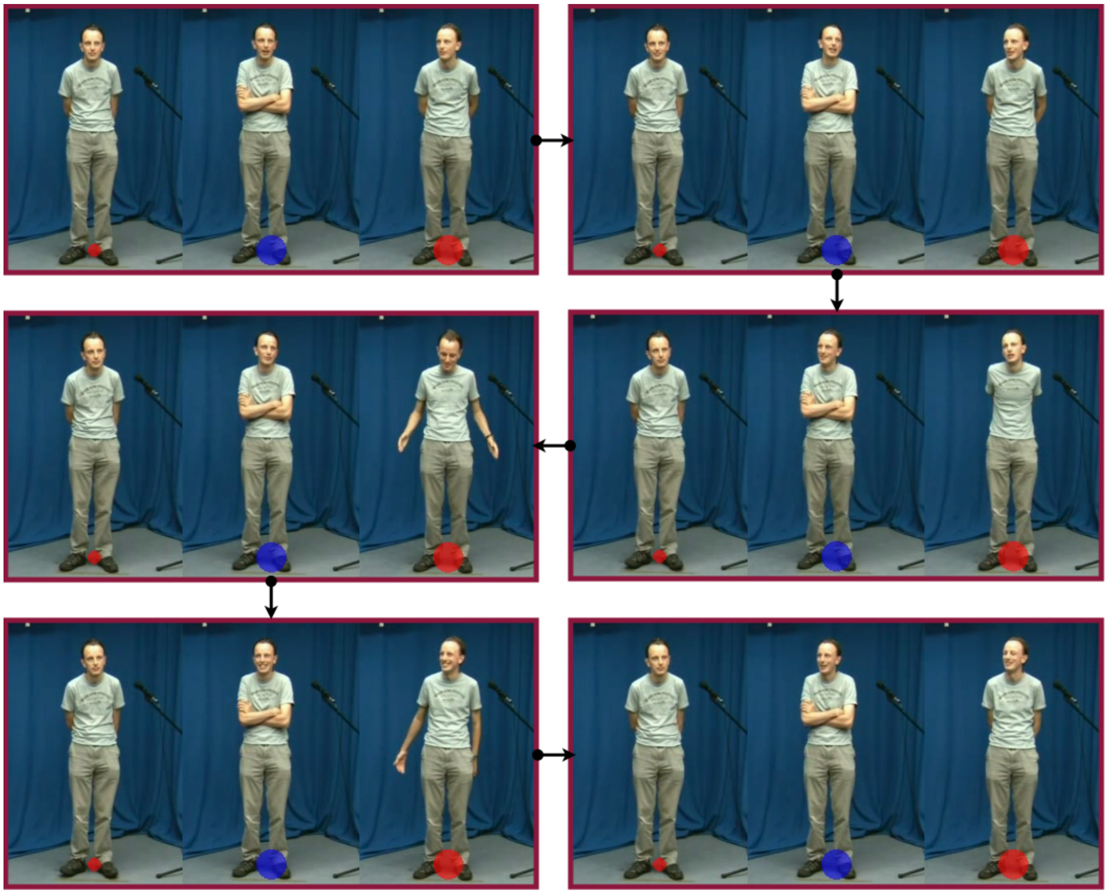


Figure 6.11: **Autonomous control of identical avatars - example 3:** Autonomous control of identical avatars of person A.

When person C is *interested*, they are highly active in the conversation, expressing several gestures and even attempting to turn-take. When person C is switched to *not interested*, their expressiveness and social activities are reduced.

A similar behaviour is displayed when using a different avatar. In Figure 6.10, person A is listening to person C. When person A is *interested* in the conversation, again they are more active, gazing more at the speaker and performing more social signals. However, when person A is *not interested*, gaze at the speaker, responsiveness and social actives are reduced.

In Figure 6.11, the diversity of the approach is demonstrated. Here, identical avatars of person A are used, with one as the speaker and two as listeners. One of the listener is *interested* in the conversation, and the other is *not interested*. The social dynamics of person A interacting with himself is not available. Instead, the social dynamic of person A, as well as person B and C are used to animate the avatar of person A, creating the illusion of person A speaking to himself. In this example, the avatar at the centre is using the dynamics of person B, the avatar on the left is using the dynamic of person A, and the avatar on the right is using the dynamics of person C. Here, the *interested* listener is better engaged in the conversation than the *not interested* listener.

To further evaluate the results and to deduce how well these animations behaved appropriately, 10 people were asked to score from 1 to 10 (1 being very low interest and 10 being very high interest), on the non-verbal behaviour of the resulting listening avatars that were animated. The subjects were unaware of the interest level of these animated listening avatars. Their scores were normalised and averaged and are presented in Table 6.2. Column 2 are the scores given to the animated listening avatars when set to *interested*, and column 3 are the scores when the animated avatars were *not interested*. The ratios of these scores are presented on the forth column. As can be seen, all scores are greater than 1 suggesting that all subjects believed the autonomous non-verbal behaviour of the avatars to appropriately matched their respective social context. A statistical significance pair t-test was also performed on the scores on Table 6.2. With *p-value* < 0.001, the null hypothesis is rejected such that there is sufficient evidence to suggest there is a difference in means across both scores when also consider-

Person	Interested	Not Interested	$\frac{\textbf{Int}}{\textbf{NotInt}}$
1	1.4	0.6	**2.3**
2	1.4	0.5	**2.8**
3	1.4	0.6	**2.3**
4	1.4	0.6	**2.3**
5	1.5	0.4	**3.5**
6	1.4	0.6	**2.3**
7	1.3	0.6	**2.1**
8	1.1	0.9	**1.2**
9	1.5	0.5	**2.7**
10	1.6	0.4	**4.4**

Table 6.2: **Scores for 10 people's perception of the animated listening avatar's *interest*:** Column 1 is the numerical index of people giving scores. Column 2 and 3 are the normalised and averaged scores for the perceived animated listening avatar's *interest* in their respective context. Column 4 is the perceived *interested* scores divided by the perceived *not interested* scores

ing their standard deviation. This proves that both the SDM and the motion synthesis approach have successfully modelled and animated appropriate social behaviour for the respective social context.

As shown in these results, the user has interactive control over who speaks, and the level of interests of the listeners. With these set parameters, the avatars interact appropriate, traversing the texture motion graph to attach video subsequences together, guided by the data mined conditional weights.

6.6 Summary

The SDM is able to derive trends between an *interested* and *not interested* listener in a conversation. These trends are successfully parameterised using data mining to derive the conditional probability of a listener's behaviour given a speaker's social signal. Human video motion modelling using a texture motion graph produces plausible transitions between cut points, allowing interactive control over a video avatar's social behaviour. Utilising the social dynamics model to drive the animation, the user can alter the interest level of participants in the conversation, effectively changing their social responses.

With relation to the research question proposed in the introduction to this thesis in Chapter 1, '*can a social model be used to drive a motion synthesis model, and generate realistic autonomous social behaviour in animations*'. Inevitably, this chapter has proven this to be true. We find that human subjects can distinguish between an *interested* and *not interested* avatar, animated autonomously using the SDM. A paired t-test further demonstrates that users were clearly able to distinguish between the two interest levels when animated, and the animated social behaviour of the avatars were sufficient for such deductions. To this end, work in this thesis contributes a novel motion model capable of animating precise human video temporal textures, and a data driven social model based on apriori association rule mining, to extract social relevant information and animate autonomous avatars.

Chapter 7

Discussion and Future Work

7.1 Motion Synthesis Models

This thesis investigated models for both a generative and example-based method for motion synthesis. Unlike other approaches, these models are easy to use, requiring the minimum amount of user intervention in creating real-time animations of motion data in various formats.

A *generative motion model* was proposed in Chapter 3, that blends between different *probability density functions* for novel motion synthesis and blending. This approach introduced a fast Gaussian approximation method based on *kd-trees* for real-time animation, combined with a Gaussian noise process to include naturalistic motion variations. Animation and motion control is real-time and interactive, and this method is also generic in data formats. The main contributions of this approach extends work by Pullen and Bregler [100], using linear blending of PDF for motion transitions, and a fast Gaussian approximation for real-time animation. A preliminary version of this work was published in [88].

Generative approaches to motion synthesis are best suited to synthesising and blending cyclic MoCap and stochastic videos, where the inconsistencies in the generalisation are less evident. When dealing with acyclic MoCap and precise textures, an example-based method is more suitable, since the original data is retained, resulting in no loss of detail.

An example-based method, MIMiC (*multimodal interactive motion controller*), was developed in Chapter 4, that can generate novel motion sequences, giving the user real-time control. The *Motion Model* can be applied to various motion formats such as 3D motion capture, video textures, and 2D tracked points. It can also produce novel sequences with the same realism inherent in the original data. The *Multimodal Controller* can provide interactive control, using a number of interfaces including audio. It is also possible to learn a conversational cue model using MIMiC to derive appropriate facial responses using audio features. This exemplar approach to motion modelling expands on previous work in *motion graph* [70, 101, 111, 13, 17, 50], using a semi-supervised approach for extracting transition points, as well as providing a multimodal and interactive means of motion control. Earlier versions of this work was published in [89, 90].

The MIMiC system incorporates a novel approach to identifying data transition points using a *k-medoid clustering algorithm.* Common methods compute an L2 distance between every frame in the sequence to derive transition points. This would be a tedious process for large data set. However, using the approach in MIMiC, by only computing the L2 distance at automatically derived k-medoid points, the amount of computation required is greatly reduced.

A limitation to MIMiC is that, both the number of k-medoid points N_c and the distance threshold of cut-point neighbours θ, are not automatic but empirically determined by the user. N_c effectively dictates the number of states/clusters in a given motion sequence. If it is set too high, it would result in unnatural animations, and if it is set too low, it would reduce the novelty of the animation and responsiveness of the controller. The minimum distance between poses/frames in a state/cluster is determined by θ. If it is too high, it becomes difficult to produce realistic blends, and if it is too low, it limits the number of possible transitions to only overlapping poses/frames in eigenspace.

Automatically deriving N_c and θ would be ideal, further reducing the amount of user intervention in the approach. A possible solution would be to incorporate a *qualitative* analysis phase to MIMIC. The *qualitative* analysis phase would firstly derive a range for N_c and θ, either selected randomly or specified by the user. For every combination of N_c and θ within this range, the average distance between neighbouring cut points are plotted against the average response time for transitioning from one motion type to another. This should produce a range of results, from having a fast average response time but with a large average distance between cut point members, to having a slow average response time but with a small average distance between cut points. The best N_c and θ combination that exists in the centre of this range can be considered as the best trade off between quality and responsiveness. However, depending on the data set and the user's requirements, manual control of N_c and θ will still be needed in order to regulate the level of connectivity and quality of the animation to satisfy the user's preference.

In the MIMiC system, although θ provides a tolerance on how close cut points need to be to form valid transitions, this does not eliminate the risk of falsely identifying transition points. Such a risk is more prominent in the motion capture data set with regards to mirrored poses (during the crossing of the legs), whereby transitions to a mirror pose will result in unrealistic reverse motion. To account for this, although MIMiC already incorporates temporal information using a 1^{st} order dynamics model, a possible improvement would be to explore the use of a 2^{nd} order dynamics to account for such ambiguities. Incorporating 2^{nd} order dynamics would take into account the start and destination states, further enforcing that only transitions that have occurred in the original data are possible. It is important to mention that, although the risk of mirrored poses exist, the applied threshold θ was sufficient to prevent its occurrence.

A largely unexplored problem is in providing autonomous control of socially interactive animations. Such a framework has vast applications which spans from computer animation, to HCI, and social science. With the means of modelling social and non-verbal behaviour, animators can more easily produce animations that look and behave convincingly. A social model can enhance man and machine interfaces, providing machines with a better understanding of natural human social behaviour, and allowing synthetic human-computer interactions to appear more like human-human interactions. Additionally, with a tool for discerning social behaviour, it may be possible to better assist social scientists with diagnosing and understanding social related conditions such as early signs of autism in children.

7.2 Social Analysis and Social Avatars

A *Social Dynamics Model* (SDM) was developed in Chapter 5, that can accurately predict conversation interest in less that 5 minutes of observation. Association rule data mined *confidence* values are used to discern trends in exchanges in social signals, without the need for psychological observations. A means for efficiently visualising the *confidence* values and their respective rules was also presented, allowing for rapid observation and derivation. Unlike previous work which use *texture* features for predictions [98, 34], our approach uses the *confidence* value of mined association rules which are both tangible (data driven) and applicable to a larger range of social contexts. An initial version of this work was published in [92].

In this work, the SDM was trained and demonstrated on a small data set. Although the objective was to model the social dynamics between the 3 subjects for the purpose of driving animations of their avatars, a larger data set is required to expand analysis for more comprehensive interpretations. Observing the dynamics of a different group of 3 participants is a possibility, cross referencing results to validate consistency. Another possibility is to use the data in the TUM AVIC interest corpus [109]. However, their annotations are subject to the opinions of 4 separate individuals, who were not participants in the conversation, as opposed to the participants themselves. As such, their opinions may be conflicting.

Visualising the SDMs were only possible for single antecedent rules. Extending the visual-SDM to view more antecedent would require a means of visualising higher order complex graphs. A possible tool is presented in [58], which present *hierarchical edge Bundles* for visualising relationships in highly complex hierarchical data. However, there is no certainty that such a visual medium would produce visually discernible trends.

By combining the SDM with MIMiC, generating photorealistic socially interactive avatars was possible (as detailed in Chapter 6). Utilising the SDM to drive the animation, the user can alter the interest level of participants in the conversation, effectively changing their social responses. Previous work in animating conversational avatars based on data driven learning methods [47, 87], differ from that presented in this thesis in that, only the *confidence* values from mined association rules are used to generate autonomous social responses. MIMiC is further extended to incorporate a *texture motion graph*, to overcome ambiguities in transitioning between video frames of natural human social behaviour. By pruning the cluster of cut points to have the same *gaze*, *talking* and *laughing* labels as their respective k-medoid centre, and by defining a finite set of strongly connected subgraphs for each unique set of social signals, the graph is made more responsive and best suited for animating natural human social behaviour. The preliminary version of this work has been published in [91].

Switching between different subsequences during photorealistic human video synthesis results in noticeable jerks. Simple blending techniques such as linear interpolation is suitable for some domains but not for precise textures in video. An intuitive video blending strategies is required. A possible approach would be to segment the subject from the background and use sub-pixel interpolation to align consecutive frames. For more drastic cases when limps are out of sync, a solution would be to detach the limps from the torso, and use some form of non-linear transformation to realistically blend between discrete frames. In this case, in-painting would also be required to fill in pixels that were occluded. This is similar to the work done in [44], except they demonstrate such an approach only on periodic hand movements.

When blending between facial expressions, a facial model can be applied to learn the dynamics of facial expressions similar to [23, 43]. Another alternative would be to generate a novel face in between frames that adheres to the transformation of the facial dynamics. A similar idea was presented in [81] but applied to static faces with no temporal coherency.

7.3 Future Work

For future work, the method applied for modelling conversational interest in the SDM can be applied to other domains. A possible expansion would be to provide real-time social behaviour analysis for multiple subjects. Given a live video feed of a crowd of people, such as a busy public park or shopping mall. From just a short period of observation, the model could determine groups of people exhibiting suspicious or antisocial behaviour. This would be a useful tool for security and surveillance personnel, making the job of monitoring a large crowd of people easier.

Another possibility could be to use the SDM as a tool in assisting sociologists and psychologists with diagnosing certain social related conditions, such early signs of autism in children. By simply extracting social relevant information from videos of children play amongst themselves, sociologists and psychologists can be better equipped in identify key behavioural trends before making diagnosis.

Companies who are heavily involved in providing customer services, could also use the SDM as a means of training staff on how to handle difficult encounters with unsatisfied and aggressive clients. A further extension could be in providing employees with real-time evaluation of their interaction with a client, in order to better cater to their needs. This is especially useful for cases where there is no face to face interaction, such as over the phone, where their is an absence of visual aid. Further study is needed to determine if speech characteristic alone can be efficiently used in predicting social context.

With the means for using a social model for understanding social context on-the-fly, the natural progression would be to enhance the socially interactive avatars to interact with real people in real-time. The avatar should be full-body and be capable of performing all social signals and whilst in different postures. As well as dialogue understanding, the system should also be able to speak and hold a conversation with multiple subjects simultaneously. As suggested earlier, a separate facial and body motion model may be a more practical approach than training a model for both, requiring less data and providing more variations. Although, an intuitive approach is required to accurately stitch both face and body parts together in the final rendering.

Dialogue understanding is a very complex problem, and additional study is needed to determine how an avatar can contribute in a natural human conversation. A possibility would be to train a model for all possible responses to a question on a specific topic, although, this may still result in artificial responses since there is only a contextual understanding of the conversation.

This real-time interactive virtual avatar would be a small step towards a true virtual companion. The ability of the user and the avatar to get on well, will only be determined by their compatibility, just like real human-human encounters. Such a system could be used in the public sector in places like train stations and airports, to provide information and assistance to passengers. It could also be used by the movie and gaming industries, for filming animated characters with human actors, or in allowing the interaction between a gamer and a game character to be more natural, enjoyable and affective.

Bibliography

[1] A. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proceedings of the ACM International Conference on Management of Data SIGMOD'93, 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In Proceedings of 20th International Conference on Very Large Data Bases, pages 487–499, 1994.

[3] A. Ahmed, F. Mokhtarian, and A. Hilton. Parametric motion blending through wavelet analysis. In Eurographics 01. Short Presentations Proceedings of Eurographics 2001., pages 347–353, 2001.

[4] M. Alexa and W. Muller. Representing animations by principal components. In Computer Graphics Forum, 19(3), pages 411–418, 2000.

[5] N. Ambady, F.J. Bernieri, and J.A. Richeson. Toward a histology of social behavior: Judgmental accuracy from thin slices of the behavioral stream. Advances in experimental social psychology, 32:201–257, 2000.

[6] N. Ambady and R. Rosenthal. Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis. Psychological Bulletin, 111(2):256–274, 1992.

[7] O. Aran and D. Gatica-Perez. Fusing audio-visual nonverbal cues to detect dominant people in conversations. In Proceedings of the 20th International Conference on Pattern Recognition, 2010.

[8] M. Argyle. The psychology of interpersonal behaviour. In Penguin, 1967.

[9] M. Argyle. Bodily communication. In Methuen, 1987.

[10] M. Argyle and M. Cook. Gaze and mutual gaze. In Cambridge University Press, 1976.

[11] O. Arikan, D.A. Forsyth, and J.F. O'Brien. Motion synthesis from annotation. In ACM Transaction on Graphics, 22, 3, July, (SIGGRAPH 2003), pages 402–408, 2003.

[12] G. Ashraf and K.C. Wong. Constrained framespace interpolation. In Computer Animation 2001, pages 61–72, 2001.

[13] K. Balci and L. Akarun. Generating motion graphs from clusters of individual poses. 24th Int. Symposium on Computer and Information Sciences, pages 436–441, 2009.

[14] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Texture mixing and texture movie synthesis using statistical learning. IEEE Transactions on Visualization and Computer Graphics, pages 120–135, 2002.

[15] A. Basharat and M. Shah. Time series prediction by chaotic modeling of nonlinear dynamical systems. In IEEE 12th International Conference on Computer Vision, pages 1941–1948. IEEE, 2009.

[16] J.B. Bavelas. Appreciating face-to-face dialogue. In AVSP 2005, 2005.

[17] P. Beaudoin, S. Coros, M. van de Panne, and P. Poulin. Motion-motif graphs. In In Proc. of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 117–126, 2008.

[18] K.S. Bhat, S.M. Seitz, J.K. Hodgins, and P.K. Khosla. Flow-based video synthesis and editing. In ACM Transactions on Graphics, SIGGRAPH 2004, 2004.

[19] N. Bianchi-Berthouze, P. Cairns, A. Cox, C. Jennett, and W.W. Kim. On posture as a modality for expressing and recognizing emotions. In Emotion and HCI workshop at BCS HCI London. Citeseer, 2006.

[20] N. Bianchi-Berthouze, W. Kim, and D. Patel. Does body movement engage you more in digital game play? And Why? Affective Computing and Intelligent Interaction, pages 102–113, 2007.

[21] Bobby Bodenheimer, Anna V. Shleyfman, and Jessica K. Hodgins. The effects of noise on the perception of animated human running. In Computer Animation and Simulation '99, pages 53–63, 1999.

[22] R. Bowden. Learning statistical models of human motion, 2000.

[23] M. Brand. Voice puppetry. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH 1999, pages 21–28. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1999.

[24] M. Brand and A. Hertzmann. Style machine. In Proceedings of SIGGRAPH, 2000, pages 183–192, 2000.

[25] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pages 353–360. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1997.

[26] Armin Bruderlin and Lance Williams. Motion signal processing. Computer Graphics, 29(Annual Conference Series):97–104, 1995.

[27] T.W. Calvert, J. Chapman, and A. Patla. Aspect of the kinematic simulation of human movement. In IEEE Computer Graphics and Applications, November 1982, volume Vol 2, No. 9, pages 41–50, 1982.

[28] S. R. Carvalho, R. Boulic, and D. Thalmann. Interactive low-dimensional human motion synthesis by combining motion models and pik. In Computer Animation and Virtual Worlds (in press), 2007.

[29] J. Cassell and H. Vilhjálmsson. Fully embodied conversational avatars: Making communicative behaviors autonomous. Autonomous Agents and Multi-Agent Systems, 2(1):45–64, 1999.

[30] Hyun-Sook Chung and Yilbyung Lee. Mcml: motion capture markup language for integration of heterogeneous motion capture data. Computer Standards and Interfaces, 26:113–130, 2004.

[31] A.D. Cliff and J.K. Ord. Space-time modeling with an application to regional forecasting. Trans, Inst. British Geographers, pages 119–128, 1975.

[32] M.F. Cohen. Interactive spacetime control for animation. ACM SIGGRAPH Computer Graphics, 26(2):293–302, 1992.

[33] S. Cooper, A. Hertzmann, and Z.Popović. Active learning for real-time motion controllers. In ACM Transaction on Graphics, 26, 3, (SIGGRAPH 2007), 2007.

[34] J.R. Curhan and A. Pentland. Thin slices of negotiation: Predicting outcomes from conversational dynamics within the first 5 minutes. Journal of Applied Psychology, 92(3):802–811, 2007.

[35] J.S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pages 361–368. ACM Press/Addison-Wesley Publishing Co., 1997.

[36] A. Dielmann and S. Renals. Automatic meeting segmentation using dynamic bayesian networks. IEEE Transactions on Multimedia, 9(1):25–36, 2007.

[37] G. Doherty-Sneddon, V. Bruce, L. Bonner, S. Longbotham, and C. Doyle. Development of gaze aversion as disengagement from visual information. In Developmental Psychology, pages 38:438–445, 2002.

[38] G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto. Dynamic textures. International Journal of Computer Vision, pages 91–109, 2003.

[39] G. Doretto and S. Soatto. Editable dynamic textures. In Proceedings of theIEEE Computer Society Conference on Computer Vision and Pattern Recognition., volume 2. IEEE, 2003.

[40] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. Personal and Ubiquitous Computing, 10(4):255–268, 2006.

[41] A. Efros and T. Leung. Texture synthesis by non-paramteric sampling. In International Conference on Computer Vision, pages 1033–1038, 1999.

[42] P. Ekman and W. Friesen. Facial action coding system. In Consulting Psychologists Press, Palo Alto, CA, 1977.

[43] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animation. ACM Transactions on Graphics (TOG), 21:388–398, 2002.

[44] M. Flagg, A. Nakazawa, Q. Zhang, S.B. Kang, Y.K. Ryu, I. Essa, and J.M. Rehg. Human video textures. In Proceedings of the 2009 symposium on Interactive 3D graphics and games, pages 199–206. ACM, 2009.

[45] A. Galata, N. Johnson, and D. Hogg. Learning structured behaviour models using variable length Markov models. In Proceedings of the IEEE International Workshop on Modelling People, 1999., pages 95–102. IEEE, 2002.

[46] D. Gatica-Perez, I. McCowan, D. Zhang, and S. Bengio. Detecting group interest-level in meetings. In Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). Citeseer, 2005.

[47] M. Gillies, X. Pan, M. Slater, and J. Shawe-Taylor. Responsive listening behavior. Computer animation and virtual worlds, 19(5):579–589, 2008.

[48] M. Gleicher. Motion editing with spacetime constraints. In Proceedings of the 1997 symposium on Interactive 3D graphics, page 139. ACM, 1997.

[49] M. Gleicher. Retargetting motion to new characters. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques, page 42. ACM, 1998.

[50] M. Gleicher, H.J. Shin, L. Kovar, and A. Jepsen. Snap-together motion: assembling run-time animations. In Proceedings of the 2003 symposium on Interactive 3D graphics, pages 181–188. ACM, 2003.

[51] E. Goffman. Replies and responses. In Language in Society, pages 257–313, 1976.

[52] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. In Proceedings of the 2004 SIGGRAPH Conference, pages 522 – 531, 2004.

[53] S. Guo and J. Roberge. A high-level control mechanism for human locomotion based on parametric frame space interpolation. In Eurographics Computer Animation and Simulation, EGCAS'96, pages 95–107, 1996.

[54] WK Hastings. Monte Carlo sampling methods using Markov chains and their applications. Biometrikal, 57(1):97–109, 1970.

[55] E. Hatfield, J.T. Cacioppo, and R.L. Rapson. Emotional contagion. Current Directions in Psychological Science, 2(3):96–100, 1993.

[56] M. Hecht, J. De Vito, and L. Guerrero. Perspectives on non-verbal communication codes, functions and contexts. In The nonverbal communication reader, pages 201–272, 2000.

[57] D.J. Heeger and J.R. Bergen. Pyramid-based texture analysis. In Proceedings of SIGGRAPH 95, August, pages 229–238, Los Angeles, California, 1995.

[58] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. IEEE Transactions on Visualization and Computer Graphics, pages 741–748, 2006.

[59] M. Iacobini, T. Gonsalves, N.B. Berthouze, and C. Frith. Creating emotional communication with interactive artwork. In 3rd International Conference on Affective Computing and Intelligent Interaction, ACII 2009., pages 1–6. IEEE, 2009.

[60] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. 4th European Conference on Computer Vision, pages 343–356, 1996.

[61] T. Jebara and A. Pentland. Action reaction learning: Analysis and synthesis of human behaviour. In Workshop on the Interpretation of Visual Motion - Computer Vision and Pattern Recognition Conference, 1998.

[62] N. Johnson, A. Galata, and D. Hogg. The acquisition and use of interaction behaviour models. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998., pages 866–871. IEEE, 1998.

[63] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. Image and Vision Computing, 14(8):609–615, 1996.

[64] A. Kendon. Some functions of gaze-direction in social interaction. In Acta Psychologica, pages 26:22–63, 1967.

[65] A. Kendon, R.M. Harris, and M.R. Key. Organisation of behavior in face to face interaction. In The Hogue, Netherlands: Mouton, 1975.

[66] A. Kleinsmith, P.R. De Silva, and N. Bianchi-Berthouze. Cross-cultural differences in recognizing affect from body posture. Interacting with computers, 18(6):1371–1389, 2006.

[67] M. Knapp. Nonverbal communication in human interaction. In Harcourt Brace College Publishers, 1972, 1972.

[68] L. Kovar and M. Gleicher. Flexible automatic motion blending with registration curves. In Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 214–224. Eurographics Association, 2003.

[69] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. In Proceedings of ACM SIGGRAPH, Aug, 2004.

[70] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In Proceedings of ACM SIGGRAPH, 21, 3, Jul, pages 473–482, 2002.

[71] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick. Graphcut textures. In ACM Transactions on Graphics, SIGGRAPH 2003, 22, 3, pages 277–286, 2003.

[72] M. LaFrance. Nonverbal synchrony and rapport: Analysis by the cross-panel technigue. In Social Psychology Quarterly, pages 42(1):66–70, 1979.

[73] M. LaFrance. Posture mirroring and rapport. In M. Davis (Ed.), Interaction rhythms: Periodicity in communicative behavior, pages 279–298. New York: Human Sciences Press, 1982.

[74] J. Lee, J. Chai, P.S.A. Reitsma, J.K. Hodgins, and N.S. Pollard. Interactive control of avatars animated with human motion data. ACM Transactions on Graphics, 21(3):491–500, 2002.

[75] J. Lee and S.Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 39–48. ACM Press/Addison-Wesley Publishing Co., 1999.

[76] C.K. Liu and Z. Popović. Synthesis of complex dynamic character motion from simple animations. ACM Transactions on Graphics (TOG), 21(3):408–416, 2002.

[77] R.M. Maatman, J. Gratch, and S. Marsella. Natural behavior of a listening agent. In Intelligent Virtual Agents, pages 25–36. Springer, 2005.

[78] A. Mehrabian and J. Friar. Encoding of attitude by a seated communicator via posture and position cues. Consulting and Clinical Psychology, pages 33, 330–336, 1969.

[79] A. Mertins and J. Rademacher. Frequency-warping invariant features for automatic speech recognition. In 2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings, volume 5, 2006.

[80] B. Meskin and J.L. Singer. Daydreaming, reflective thought, and laterality of eye movements. In Journal of Personality and Social Psychology, pages 30:64–71, 1974.

[81] U. Mohammed, S.J.D. Prince, and J. Kautz. Visio-lization: generating novel facial images. ACM Transactions on Graphics (TOG), 28:1–8, 2009.

[82] A.W. Moore. An intoductory tutorial on kd-trees. University of Cambridge Computer Laboratory Technical Report No. 209, Extract from PhD Thesis, 1991.

[83] E. Moulines, F. Emerard, D. Larreur, JL Le Saint Milon, L. Le Faucheur, F. Marty, F. Charpentier, and C. Sorin. A real-time French text-to-speech system generating high-quality synthetic speech. In International Conference on Acoustics, Speech, and Signal Processing, ICASSP, pages 309–312. IEEE, 1990.

[84] T. Mukai and S. Kuriyama. Geostatistical motion interpolation. In ACM Transaction on Graphics, 24, 3, Aug, pages 1071–1081, 2005.

[85] A. Mulac, K. Erlandson, W.J. Farrar, J.S. Hallett, J.L. Molloy, and M.E. Prescott. 'uh-huh. what's that all about?' differing interpretations of conversational backchannels and questions as sources of miscommunication across gender boundaries. In Communication Research, pages 25:641–669, 1998.

[86] Eadweard Muybridge. Animals in motion. In Dover Publications, 1974.

[87] R. Niewiadomski, S. Hyniewska, and C. Pelachaud. Modeling emotional expressions as sequences of behaviors. In Proceedings of the 7th International Conference on Intelligent Virtual Agents (IVA), 2009.

[88] D. Okwechime and R. Bowden. A generative model for motion synthesis and blending using probability density estimation. In Fifth Conference on Articulated Motion and Deformable Objects, 9-11 July, Mallorca, Spain, 2008.

[89] D. Okwechime, E. J. Ong, and R. Bowden. Real-time motion control using pose space probability density estimation. In IEEE Int. Workshop on Human-Computer Interaction, 2009.

[90] D. Okwechime, E. J. Ong, and R. Bowden. Mimic: Multimodal interactive motion controller. In IEEE Transactions on Multimedia, 2011.

[91] D. Okwechime, E-J. Ong, A. Gilbert, and R. Bowden. Social Interactive Human Video Synthesis. In Tenth Asian Conference on Computer Vision. Springer, 2010.

[92] D. Okwechime, E. J. Ong, A. Gilbert, and R. Bowden. Visualisation and prediction of conversation interest through mined social signals. In IEEE Int. Workshop on Social Behavior Analysis, 2011.

[93] E. J. Ong, Y. Lan, B. J. Theobald, R. Harvey, and R. Bowden. Robust facial feature tracking using selected multi-resolution linear predictors. In Int. Conf. Computer Vision ICCV 2009, 2009.

[94] R. Paget and I.D. Longstaff. Texture synthesis via a noncausal nonparametric multiscale Markov random field. Image Processing, IEEE Transactions on, pages 925–931, 2002.

[95] C.J. Park, I.K. Jeong, H.K. Kim, and K. Wohn. Sensor fusion for motion capture system based on system identification. In IEEE Computer Animation, pages 82–86, 2000.

[96] S.I. Park, H.J. Shin, and S.Y. Shin. On-line locomotion generation based on motion blending. In Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 105–111. ACM, 2002.

[97] A. Pentland. A computational model of social signaling. In 18th Int. Conf. on Pattern Recognition. ICPR, 2006.

[98] A. Pentland. Social signal processing. In IEEE Signal Processing Magazine, pages 108–111, 2007.

[99] Z. Popović and A. Witkin. Physically based motion transformation. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 11–20. ACM Press/Addison-Wesley Publishing Co., 1999.

[100] K. Pullen and C. Bregler. Synthesis of cyclic motions with texture, 2002.

[101] H. Rachel and M. Gleicher. Parametric motion graph. 24th Int. Symposium on Interactive 3D Graphics and Games, pages 129–136, 2007.

[102] C. Rose, M.F. Cohen, and B.Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. In IEEE Computer Graphics and Applications, September/October, pages 32–40, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[103] C. Rose, P.P.J. Sloan, and M.F. Cohen. Artist-Directed Inverse-Kinematics Using Radial Basis Function Interpolation. In Eurographics, pages 239–250. Wiley Online Library, 2001.

[104] E. Sahouria and A.Zakhor. Content analysis of video using principal components. In IEEE Trans. on Circuits and Systems for Video Technology, 1999.

[105] A. Schödl and I.A. Essa. Controlled animation of video sprites. In Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 121–127. ACM, 2002.

[106] A. Schödl, R. Szeliski, D.H. Salesin, and I. Essa. Video textures. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH 2000, pages 489–498. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 2000.

[107] M. Schroder, E. Bevacqua, F. Eyben, H. Gunes, D. Heylen, M. ter Maat, S. Pammi, M. Pantic, C. Pelachaud, B. Schuller, et al. A demonstration of audiovisual sensitive artificial listeners. In 3rd International Conference on Affective Computing and Intelligent Interaction, ACII 2009., pages 1–2. IEEE, 2009.

[108] M. Schröder, D. Heylen, and I. Poggi. Preception of non-verbal emotional listener feedback. In Proceedings of Speech Prosody, 2006.

[109] B. Schuller, R. Müller, F. Eyben, J. Gast, B. Hörnler, M. Wöllmer, G. Rigoll, A. Höthker, and H. Konosu. Being bored? Recognising natural interest by extensive audiovisual integration for real-life application. Image and Vision Computing, 27(12):1760–1774, 2009.

[110] B. Schuller and G. Rigoll. Recognising interest in conversational speech–comparing bag of frames and supra-segmental features. In Proc. InterSpeech, pages 1999–2002, 2009.

[111] H.J. Shin and H.S. Oh. Fat graphs: Constructing an interactive character with continuous controls. In Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, page 298, 2006.

[112] P.P. Sloan, C. Rose, and M.F. Cohen. Shape and animation by example. Technical Report MSR-TR-2000-79, 2000.

[113] P.P.J. Sloan, C. Rose, and M.F. Cohen. Shape by example. In Proceedings of the 2001 symposium on Interactive 3D graphics, pages 135–143. ACM, 2001.

[114] S. Soatto, G. Doretto, and Y.N. Wu. Dynamic textures. In Eighth IEEE International Conference on Computer Vision, volume 2, pages 439–446. IEEE, 2002.

[115] M. Stone, D. DeCarlo, I. Oh, C. Rodriguez, A. Stere, A. Lees, and C. Bregler. Speaking with hands: Creating animated conversational characters from recordings of human performance. In ACM Transactions on Graphics (TOG), SIGGRAPH 2004, volume 23, pages 506–513. ACM New York, NY, USA, 2004.

[116] S. Sudarsky and D. House. Motion capture data manipulation and reuse via B-splines. CAPTECH'98, pages 55–69, 1998.

[117] S. Sudarsky and D. House. An integrated approach towards the representation, manipulation and reuse of pre-recorded motion. IEEE Computer Animation Conference, page 6571, 2000.

[118] W. Sun and De Montfort University. Modelling bipedal locomotion using wavelets for figure animation. De Montfort University, 2000.

[119] M. Szummer and R. Picard. Temporal texture modeling. In In Proceeding of IEEE International Conference on Image Processing, 1996, pages 823–826, 1996.

[120] L. Molina Tanco and A. Hilton. Realistic synthesis of novel human movements from a database of motion captured examples. In Proceedings of the IEE Workshop on Human Motion HUMO 2000), 2000.

[121] R. Tarjan. Depth first search and linear graph algorithm. SIAM Journal of Computing 1, pages 146–160, 1972.

[122] L. Torresani, P. Hackney, and C. Bregler. Learning motion style synthesis from perceptual observation. In Proceedings Neural Information Processing Systems Foundation NIPS 19, 2007.

[123] A. Treuille, Y. Lee, and Z. Popovic. Near-optimal character animation with continuous control. In Proceedings of SIGGRAPH 2007 26(3), 2007.

[124] Nikolaus F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. J. Vis., 2(5):371–387, 9 2002.

[125] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In Proceedings of the 22nd annual ACM conference on Computer graphics, 06-11 August, pages 91–96, Los Angeles, California, 1995. Addison Wesley.

[126] M. Unuma and R. Takeuchi. Generation of human motion with emotion. In Computer Animation, pages 77–88, 1993.

[127] R. Vertegaal, I. Weevers, C. Sohn, and C. Cheung. Gaze-2: conveying eye contact in group videoconferencing using eye-controlled camera direction. In Proceedings of CHI 2003. ACM Press, 2003.

[128] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. Image and Vision Computing, 27(12):1743–1759, 2009.

[129] A. Vinciarelli, M. Pantic, H. Bourlard, and A. Pentland. Social signal processing: state-of-the-art and future perspectives of an emerging domain. In Proceeding of the 16th ACM international conference on Multimedia, pages 1061–1070. ACM, 2008.

[130] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In Proc. IEEE CVPR 2001, 2001.

[131] A. Waibel, T. Schultz, M. Bett, M. Denecke, R. Malkin, I. Rogina, R. Stiefelhagen, and J. Yang. SMaRT: The smart meeting room task at ISL. In 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03), volume 4, 2003.

[132] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. IEEE transactions on pattern analysis and machine intelligence, 30(2):283–298, 2008.

[133] Y. Wang and S.C. Zhu. A generative method for textured motion: Analysis and synthesis. European Conference on Computer Vision, ECCV 2002, pages 583–598, 2002.

[134] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structure vector quantization. In Proceedings of SIGGRAPH 2000, July, pages 479–488, 2000.

[135] D.J. Wiley and J.K. Hahn. Interpolation synthesis of articulated figure motion. In IEEE Computer Graphics and Applications, November/December, pages 17(6):39–45, 1997.

[136] A. Witkin and M. Kass. Spacetime constraints. ACM Siggraph Computer Graphics, 22(4):159–168, 1988.

[137] A. Witkin and Z. Popovic. Motion warping. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, 1995, pages 105–108, NY, USA, 1995. ACM Press.

[138] V. Yngve. On getting a word in edgewise. In Papers from the sixth regional meeting of the chicago linguistic society, pages 567–577, 1970.