A Generic Framework for Editing and Synthesizing Multimodal Data with Relative Emotion Strength

Jacky C. P. Chan

Caritas Institute of Higher Education, Hong Kong

email: j2chan@cihe.edu.hk

Hubert P. H. Shum

Northumbria University, United Kingdom

email: hubert.shum@northumbria.ac.uk

He Wang

University of Leeds, United Kingdom

email: H.E.Wang@leeds.ac.uk

Li Yi

Yilifilm, Shenzhen, China

email: yilistudio@126.com

Wei Wei

Xian University of Technology, Xian, China

email: weiwei@xaut.edu.cn

Edmond S. L. Ho

Northumbria University, United Kingdom

email: e.ho@northumbria.ac.uk

Abstract

Emotion is considered to be a core element in performances [1]. In computer animation, both body motions and facial expressions are two popular mediums for a character to express the emotion. However, there has been limited research in studying how to effectively synthesize these two types of character movements using different levels of emotion strength with intuitive control, which is difficult to be modelled effectively. In this work, we explore a common model that can be used to represent the emotion for the applications of body motions and facial expressions synthesis. Unlike previous work which encode emotions into discrete motion style descriptors, we propose a continuous control indicator called *emotion strength*, by controlling which a data-driven approach is presented to synthesize motions with fine control over emotions. Rather than interpolating motion features to synthesize new motion as in existing work, our method explicitly learns a model mapping low-level motion features to the emotion strength. Since the motion synthesis model is learned in the training stage, the computation time required for synthesizing motions at run-time is very low. We further demonstrate the generality of our proposed framework by editing 2D face images using relative emotion strength. As a result, our method can be applied to interactive applications such as computer games, image editing tools and virtual reality applications, as well as offline applications such as animation and movie production.

Keywords: motion capture, data-driven, motion synthesis, emotion motion, image editing, facial expression, relative attribute

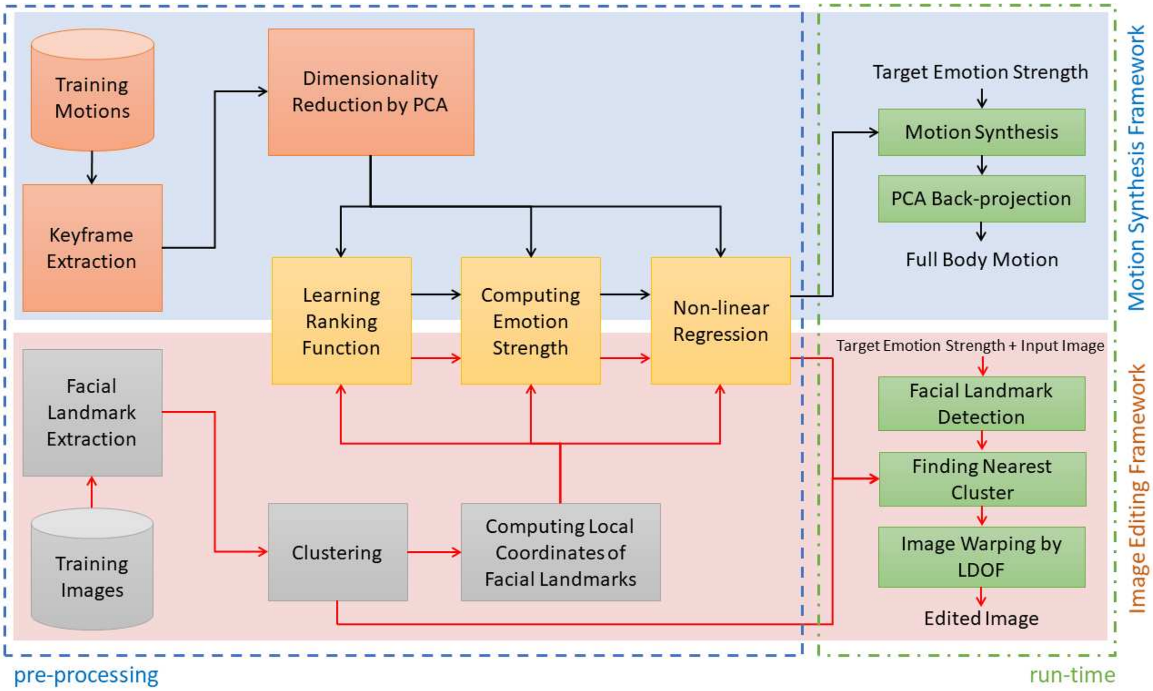


Figure 1: Overview of the proposed methodology. Both of the motion synthesizing framework (upper) and the face image editing framework are sharing the same set of core algorithms.

1 Introduction

Editing and synthesizing multimedia data such as 3D human motions and 2D images is a challenging task due to the high dimensionality of the data representation as well as the subjective aesthetic judgment. As a result, most of the tasks in human motion and image editing are still completed manually nowadays in the entertainment industry. With the advancement of machine learning algorithms and technologies, *learning how to edit and synthesize multimedia data* becomes an appealing direction.

Synthesizing realistic human motion from existing motion data, either acquired by motion capture (MOCAP) devices or hand-crafted, has been an active research area in the past two decades. In graphics applications such as movies, animation, interactive computer games and virtual reality (VR), realism of human motion plays an important role in enhancing the user's experience. In the past, motion naturalness and the ease of control have been intensively studied. While they make the character animation tasks easier, we argue that naturalness and control alone are not enough to generate realistic and diversified motions.

Emotion is considered to be a core element in performances [1], which is an indispensable piece in motion realism. In computer animation, both body motions and facial expressions are two popular mediums for a character to express the emotion. To understand and represent the emotion, there is an increasing focus on analyzing how motion emotion are perceived by users [2, 3, 4, 5]. While it is possible to identify emotion automatically from motion, synthesizing motion with controllable emotion intensity is much more challenging due to the high-dimensionality in the control parameters as well as the complex relationship between such control parameters and the emotional expressions.

The most relevant stream of work is style transfer from example motions to a new one [6, 7]. However, the existing work solves this problem by interpolating sample motions. We argue that a drawback of such an approach is *interpolating the motion but not necessarily interpolating the emotion*. In other words, existing methods mainly learn low-level motions features in the way that the control of interpolation only happens at the motion level, not at the emotion level. This is because the learning methods ignore the relations between low-level motion features and high-level emotions. In this work, we tackle this problem by learning the relationship between low-level motion features (such as joint positions, velocities) and high-level emotion attributes with different levels of strengths.

Learning the relationship between low-level motion features and mid-/high-level semantic representation has been explored in the computer vision community. *Attributes* [8] have been widely used in visual recognition tasks. In particular, relative attributes [9] that indicate the relative strength of an attribute between two samples can naturally be used to describe the differences in high-level semantics such as styles and emotions among different motions. The relationship between motion features and the attribute can be learned from the training data to facilitate motion synthesis.

In this paper, we propose a new data-driven method to synthesize motion with controlled emotion expressions. The goal is to synthesize motions with emotions intuitively, such as directly specifying the level of *happiness* (i.e a scalar scale) of the motion to be edited. Our method treats the emotion of each motion as a high-level representation (i.e. attribute) and learns the relative strength of the emotion by pairwise relative attribute. A scoring function that evaluates the attribute strength will be learned and used to guide the motion synthesis process. To our knowledge, our method is the first attempt to learn the relationship between low-level motion features and human nameable attributes for motion synthesis instead of interpolating motion directly. Since the learning process is done in the pre-processing stage, the run time computational cost is very low. Our proposed methods are general which do not make any assumptions on the features computed from the data. We further demonstrate the generality of our proposed framework by editing 2D face images using relative emotion strength. As a result, our method can be applied to interactive applications such as computer games, image editing tools and VR applications, as well as offline applications such as animation and movie production.

The main contributions of this work can be summarized as follows:

- The first method to learn the relationship between low-level features and the strength of emotion expressions for human motion and facial expression synthesis
- A real-time motion synthesis framework that is controlled solely by the strength of emotion expressions
- A real-time face image editing framework that is controlled solely by the strength of emotion expressions

2 Related Work

In this section, we will first review the related work in editing and synthesizing human motion with styles, which creates new motions based on high-level properties. Next, we review the work in learning the relationship between high-level semantic representations and low-level features from the input data. Finally, we will review some of the recent work in editing face images.

2.1 Style-based motion synthesis and editing

An early work by [10] proposed to synthesize styled motion sequences by interpolating and extrapolating the styles from the training motion data using a learned statistical model called *style machine*. [11] proposed an approach to project input motion to lower dimensional space by Principal Component Analysis (PCA) and compare the difference with other example motions in the database. The differences are then to be used as the style of the input motion and transferred to other motions. [12] proposed to learn the difference in the styles in motion pairs (e.g. a neutral and a styled motion as input and output styles). The relationship between the input and output motions in training can then be described by a linear time-invariant (LTI) model. Using such a model, the learned style can be transferred to a new input motion. [13] proposed a method for editing motion using Gaussian process models of dynamics and kinematics. Such an approach can be used for motion style transfer. [6] proposed a real-time approach for style transfer for human motion. Their method represents the style difference between the input and output motions by a time-varying mixture of autoregressive models. Their method learns such models automatically from unlabeled heterogeneous motion data.

While many style editing and transferring approaches have been proposed, most of the work focuses on finding the differences between motions with different styles and creating new motion by interpolation. We argue that such an approach does not necessarily interpolate the underlying emotion of the motion. In this work, we learn the relationship between motion features and emotion strength such that the motion synthesis process can be directly controlled using the target emotion strength.

2.2 Attribute-based representations

Visual attributes are human-nameable semantic concepts of things in the world and are popular in the computer vision community for image recognition [8]. [9] further proposed the concept of the relative attribute to capture the relative attribute strength between data. The aforementioned approaches mainly focused on handling visual recognition tasks (e.g. image classification). A recent work showed that images (e.g. 2D outdoor scenes) can be edited by applying the relative attribute concepts [14]. Another recent work learns the mapping between high-level descriptors (e.g. softness, silkiness) and simulation parameters for cloth simulation [15]. A perceptual control space is then created and the cloth simulation can be controlled by adjusting the degrees in selected high-level descriptors. These recent studies motivated us to learn such a relative attribute strength for motion synthesis.

To sum up, most of the existing work focuses on learning the attribute strength from input data for recognition tasks. While some recent work applied the concept of attribute-based editing in the computer graphics community, our work is the first to make use of the attribute strength for human motion editing.

2.3 Facial image editing

Editing facial image attracts a lot of interests in the Computer Vision community in recent years. Kaneko et al. [16] proposed a new deep learning framework for editing face image by attributes. A Generative Adversarial Network (GAN) is used for learning the distribution of the face image with emotion expression in the space. Since then, a wide range of GAN-based face image editing approaches are presented. Expression Generative Adversarial Network (ExprGAN) [17] encodes the ”expressive code” to improve the expressiveness of the image representation and results in higher image quality based on a GAN network. In Computer Graphics, a recent work proposed by Averbuch-Elor et al. [18] animates a single portrait by an input video. The driving video is captured from another subject and the proposed method extracts the facial deformation pattern from the video. The deformation pattern is then applied to the target image by 2D image warping. Additional details such as creases and wrinkles are added to the target image to further improve the quality of the resultant image. Portenier et al. [19] proposed a deep learning framework based on deep Convolution Neural Networks (CNN) and GAN for sketch-based face image editing.

While we share some similar interests in editing face image with intuitive controls as in the aforementioned work, the existing work is either 1) require additional guidance for control (such as driving video), or 2) solely focusing on editing face images. In contrast, the framework proposed in this paper is generic and can be applied to both human skeletal motion and face image editing.

3 Overview

For the sake of clarity in presenting the proposed methods, the motion synthesis framework will be explained first in Section 4. Then, the proposed face image editing framework will be presented in Section 5.

The overview of our proposed motion synthesis framework is illustrated in Figure 1. There are two stages in the proposed framework: pre-processing and run-time stages. The pre-processing stage learns models for analyzing and synthesizing motions with different emotion strength. Firstly, we collect motions with emotion labels as the input of our framework. Next, the motions are represented compactly by keyframing and dimensionality reduction. Then, the relative emotion strength of the training motion data is computed by a ranking function. Finally, the relative emotion strength and compact motion representation will be used for training a motion synthesis model.

During run-time, the target emotion strength (i.e. a scalar value) will be used as input to control how the motion will be synthesized using the model trained in the pre-processing stage. Full body motion can be obtained by back-projecting the synthesized compact motion representation to full body motion space.

4 Learning Attribute Strength

In this section, we present the core methodology of the new data-driven motion synthesis approach that learns the relationship between low-level motion features and the emotion strength. Firstly, the motion data being used in our approach for learning will be explained in Section 4.1. Next, we propose to use a compact motion representation that removes the temporal redundancy of the motion in Section 4.2. Thirdly, the dimensionality of the frame-based (i.e. pose) representation (Section 4.2.3) is further reduced to facilitate the learning process in a later stage. Finally, the learning process of the ranking function that computes the emotion strength will be explained in Section 4.3.

4.1 Motions with Emotions

In order to analyze and learn the relationship between low-level motion features and the strength of emotion expressions, a significant amount of human motion data has to be collected. In this work, the MOCAP data from the Body Movement Library [20] was used. This dataset contains 4080 human motion sequences captured using a commercial optical motion capture system. The motions were captured from 30 (15 male and 15 female, ranging from 17 to 29 years old and with a mean age of 22 years old) *non-professional participants* in order to capture natural emotion expressions rather than capturing staged/exaggerated motions from professional performers. There are three motion types: *knocking, lifting,* and *throwing*.

Each motion contains a sequence of poses captured from a single subject and represented by the joint positions in Cartesian coordinates in each frame. From the raw motion data, we extracted 33 joint positions in each frame. The duration of each motion sequence ranges from 65 to 457 frames at 60 Hz. The dataset was designed for recognizing identity, gender and emotion from the motion data. As a result, each subject performed each motion type with 4 different emotion expressions: *Neutral, Angry, Happy,* and *Sad* .

In this work, we focus on using two emotion models, namely *Happiness* and *Anger*, because a continuous emotion parameterization can be achieved. Specifically, for the *Happiness* model, we define *Happy* and *Sad* to be opposite to each other and *Neutral* to be at the middle. For the *Anger* model, we define *Anger* and *Neutral* as two extremes. As a result, we can simplify the parameterization as a single scalar value to control the emotion level continuously in *Happiness* (*Happy* $\leftrightarrow$ *Neutral* $\leftrightarrow$ *Sad*) and *Anger* (*Angry* $\leftrightarrow$ *Neutral*). This allows us to control the emotion strength continuously.

4.2 Motion Representation

Having presented the details of the motion data for training, we now explain the compact motion representation used in our proposed framework. Each motion M contains a sequence of poses p, i.e. $M = \{p_1, ..., p_n\}$, where n is the total number of frames (or poses) of M. Each pose p_i is represented by the 3D joint positions in Cartesian coordinates. As there are 33 joints extracted from the raw motion data, each pose is represented by a 99-dimensional vector. In order to facilitate the learning process, we normalize all data by removing the translation and the vertical rotation (i.e. y-axis) of the root joint in the first frame as in other data-driven approaches (e.g. [21], [22]). By normalizing the pose in this way, the differences of the postures can be compared directly in the Euclidean distance of the 3D joint positions. This has been widely used in data-driven motion synthesis approaches such as Motion Graphs [22].

4.2.1 Keyframe Extraction

Human motions are naturally temporally redundant and removing such redundancy can facilitate the the learning process in later stages. In addition, the motion sequences we collected have different durations. As a result, As a result, it is difficult to carry out machine learning from the training motion without preprocessing the data [23]. To tackle the aforementioned problems, keyframes of each motion are extracted by Curve Simplification algorithm [24]. Given a motion M, a set of q keyframes $K = \{k_1, ...k_q\}$ will be extracted, which minimizes the reconstruction error when interpolating the in-between motion using spline interpolation. In this study, we tested the reconstruction error with 10-15 keyframes and empirically found that using 13 keyframes can balance the trade-off between reconstruction error and compactness of the motion representation.

4.2.2 Joint Velocity

We observed that some of the subjects expressed different emotions by using different speed and rhythm when performing the motion. Using the *knocking door* motion as an example. The subjects tended to move faster when they were *happy*, and moved slower when they were *sad*. For this reason, the velocity of the joints between adjacent keyframes is computed:

$$v_{i+1} = \frac{k_{i+1} - k_i}{\Delta t} \tag{1}$$

where Δt is the duration between the two adjacent keyframes. Here, we compute the joint velocity from adjacent keyframes instead of adjacent frames in the original motion because we want to reconstruct the full motion into different durations by editing the velocity. Specifically, given the adjacent keyframes and the joint velocity, the duration between two adjacent keyframes can be approximated by:

$$\Delta t = \frac{k_{i+1} - k_i}{v_{i+1}} \tag{2}$$

By this, the speed of the motion can be adjusted easily using this compact representation. Finally, each keyframe contains the pose features k_i and velocity features v_i and results in a $99 + 99 = 198$-dimensional feature vector.

4.2.3 Manifold Learning

It has been argued that natural motions form a motion manifold and can be embedded in a lower dimensional space [25], which gives a compact representation of motions. To further reduce the dimensionality of the pre-frame (i.e. pose) feature to facilitate the learning process (will be explained in Section 4.3), the pose features and velocity features are concatenated and projected to a low-dimensional space. While many dimensionality reduction approaches are available, Principal Component Analysis (PCA) has been used for analyzing the style difference for motion synthesis [11]. This leads us to use such a simple, low-computational cost and widely used dimensionality reduction techniques in the proposed method.

One of the issues is to select an appropriate dimensionality in the latent space such that the essential information in the original motion is retained. We empirically calculate the reconstruction error (i.e. back-projecting the latent representation and compare the result with the keyframe features) using different numbers of dimension in the latent space. We found that projecting the keyframe features (for each keyframe) from 198-d to 40-d achieved low reconstruction error and we use this setting in all experiments.

4.3 Learning the Ranking Function

In this section, we will explain how to learn the relationship between the latent representation and the strength of the attribute from two sets of inputs: i) the compact motion features explained in Section 4.2.3 and ii) the emotion label associated with each motion. One simple way to learn such a function is to train a regression function on the ground truth emotion strength and the corresponding motion features. However, the ground truth attribute strength may not be available in the database. In the Body Movement Library database we used, only a single label (i.e. the emotion) is associated with each motion. The relative emotion strength between motions with the same class label is not available.

Inspired by [9], a ranking function can be learned from a small set of pairwise training samples with relative ranking on an attribute (i.e. emotion in our framework). Such a ranking function can then be used for computing the attribute strength of unseen data. The learning process can be formulated as a max-margin optimization problem. Specifically, we learn a ranking function $r(x_i)$ in which :

$$r(x_i) = wx_i \tag{3}$$

to weight each input feature and return the weighted sum as the attribute score that indicates the attribute strength. When solving for the ranking function, a set of *relative constraints* have to be satisfied. Using the notation in [9]:

$$\forall(i, j) \in O : wx_i > wx_j \tag{4}$$

$$\forall(i, j) \in S : wx_i = wx_j \tag{5}$$

where O and S are sets that contain *ordered* and *similar* paired samples, and x_i and x_j are the feature vectors of the i-th and j-th samples (motions in our approach).

More specifically, the ordered set O contains motions with differences in ground truth attribute strength in each pair as in Eq. 4. In the dataset we used, we setup the ordered pairwise relative constraints according to the emotion labels of the training motions: *Happy* $>$ *Neutral* $>$ *Sad* and *Angry* $>$ *Neutral* when training the *Happiness* and *Anger* ranking functions, respectively. On the other hand, the similar set S contains motions with similar ground truth attribute strength in each pair as in Eq. 5. The similar set contains the pairwise motions with the same emotion label in our proposed framework. The attribute strength of each sample can be computed by multiplying the learned weights w with the feature vector (e.g. x_i or x_j) and our task is to learn w by:

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}\|w\|_2^2 + C\Big(\sum \xi_{ij}^2 + \sum \gamma_{ij}^2\Big) \\ \text{s.t.} \quad & w(x_i - x_j) \geq 1 - \xi_{ij}; \forall (i,j) \in O \\ & |w(x_i - x_j)| \leq \gamma_{ij}; \forall (i,j) \in S \\ & \xi_{ij} \geq 0; \gamma_{ij} \geq 0, \end{aligned} \tag{6}$$

where C is the trade-off parameter to control the softness of the pairwise relative constraints to be satisfied, and ξ_{ij} and γ_{ij} are slack variables. Introducing the slack variables are serving the purpose of relaxing the constraints on classifying all training samples perfectly as in conventional SVMs. Since there are two groups of samples, namely ordered and similar, ξ_{ij} and γ_{ij} are introduced to relax the constraints for solving for w to satisfy all of the ordered and similar samples, respectively. This primal problem can be solved efficiently by Newton's method [26].

4.4 Motion Synthesis by Attribute Strength

Having learned the ranking function as explained in Section 4.3, the emotion strength (a scalar) of each training motion can be computed. Since our ultimate goal is to synthesize new motion by specifying the emotion strength, here we propose to learn a regression model on the emotion strength and the corresponding motion features from the training data. Specifically, we learn a regression function $f(s)$ that takes the target emotion strength s as input and returns the dimensionality reduced motion feature x:

$$x = f(s) \tag{7}$$

In the implementation of the proposed framework, we train the regression function using the Neural Network regression model in Matlab [27]. The implementation details will be explained in Section 6.1.

5 Editing Facial Expression on 2D Images by Attribute Strength

Having presented the method to learn the mapping between high-level attribute strength and low-level motion features for motion editing tasks in previous sections, we further show the generality of our proposed method by editing facial expressions on 2D images. In particular, the process to the ranking function is exactly the same as in the proposed motion editing framework (Section 4.3). The main difference is the features used in the training. The details will be explained below.

5.1 Face Images and Pre-processing

The 2D images used in the training process and experiments are obtained from the Labeled Faces in the Wild (LFW) [28] dataset. The dataset contains 13233 images captured from 5749 subjects. The attributes of each image are calculated using the method proposed by

Kumar et al. [29] and those values are available to the public on the author's website.

Since the face images are obtained from different sources (so-called 'in the wild'), image pre-processing is required to facilitate the learning process in the later stages in the framework. In particular, the illumination variations in 2D images will make the learning process more challenging. To tackle this problem, 2D facial landmark positions are used in our proposed framework as there will be no color and image appearance information will be included in the training process. Using 2D locations can keep the facial expression representation abstract to facilitate the attribute strength learning process as well.

In the pre-processing stage, the facial landmarks on each image are detected using the method proposed by Zhu and Ramanan [30]. The source code provided by the authors [30] were used in our experiments and 66 facial landmarks are extracted from each image. The facial landmarks detected from two examples in the LFW dataset are illustrated in Figure 2. The facial landmark positions are represented by a set of 2D coordinates.

5.2 Handling Pose Variation

In addition to the illumination variations, pose variations is another problem which affects the performance of the learning process when images collected/captured from different sources. To alleviate this problem, we cluster the images based on the contour of the face. Specifically, the 2D facial landmarks coordinates are converted into local coordinates using the average position of all extracted landmarks as the origin. Then, all images are clustered

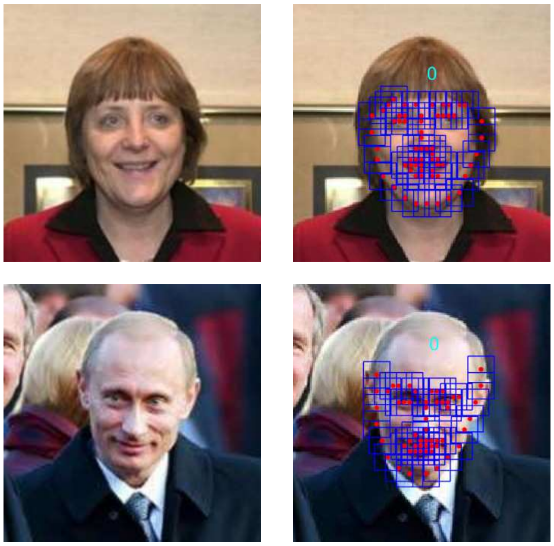


Figure 2: 68 facial landmarks (red dots) are detected using the method proposed by Zhu and Ramanan [30].

by the local coordinates of the landmarks located on the contour. In our experiment, we found that using k-means clustering with $k = 12$ to cluster the faces can lead to reasonable cluster sizes for the learning tasks while containing faces with similar poses within each cluster. An example is shown in Figure 3 and the centroids clearly show the pose variations in the dataset we used.

5.3 Facial Expression Representation

While human faces have the same structure, the distances between different regions (i.e. eyes, nose and mouth) vary on different faces. Directly using the absolute 2D facial landmark positions extracted using [30] will produce sub-optimal results. To tackle this problem, the absolute facial landmark coordinates are converted into the local coordinates within each

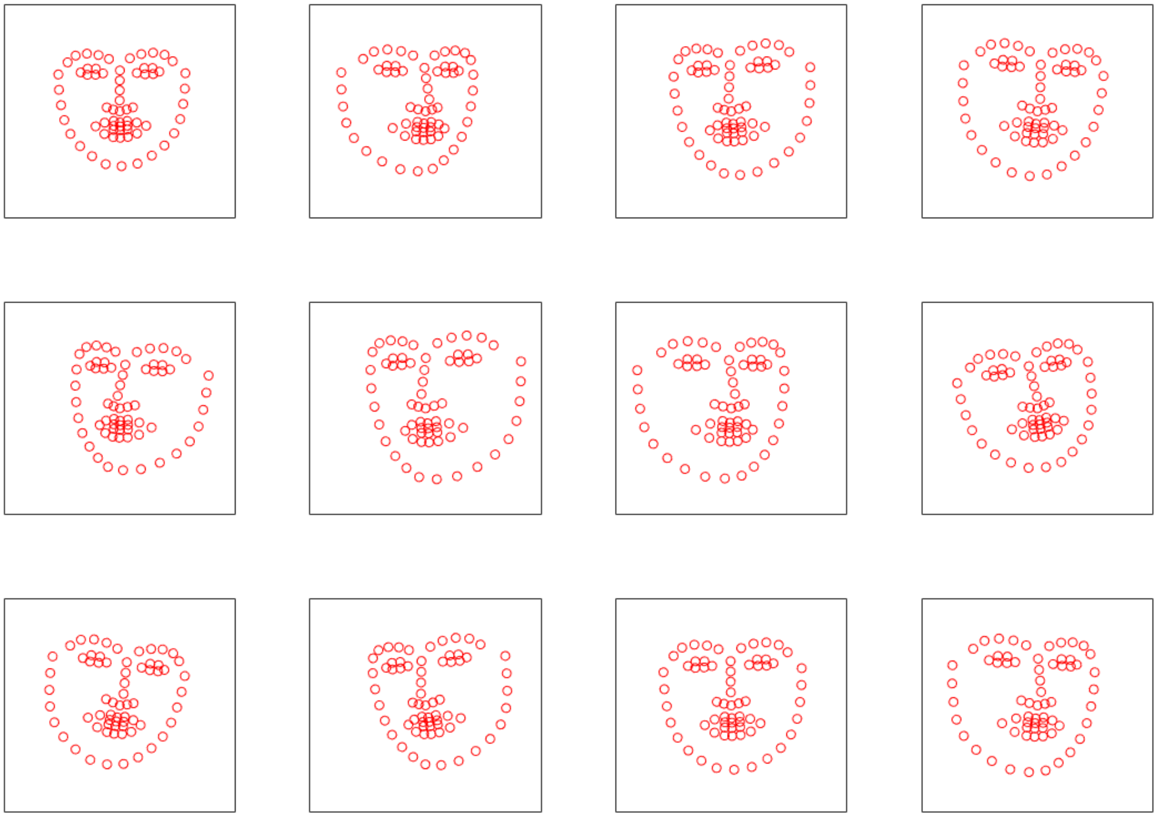


Figure 3: An example of the facial landmark positions of the 12 centroids computed from the clustering process.

region. The origin of each region is defined by the average position of all the landmarks belonging to that region. Finally, we concatenate the local 2D coordinates from 4 regions (two eyes, nose and mouth) into a single vector as the input in the attribute strength learning process. Each face is represented by 49 landmarks and the 98-d vector contains the corresponding local coordinates.

5.4 Learning Attribute Strength on Facial Features

The same ranking function learning process explained in Section 4.3 is used for learning the attribute strength from 2D images. To handle the pose variations, the ranking function is learned in each of the 12 face image clusters. The 'smile' attribute values computed by the attribute prediction method [29] are used to construct O and S (the sets that contain *ordered* and *similar* paired samples, explained in Section 4.3) for the training process. In particular, we define the happiness attribute strength as *the higher the smile attribute value, the more happy the subject is*.

5.5 2D Image Editing by Attribute Strength

Having learned the ranking function as explained in Section 5.4, the emotion strength, which is a scalar, of each training facial landmark feature can be computed. Again, we learn a regression model on the emotion strength and the corresponding facial landmark features from the training data as in the motion synthesis framework explained in Section 4.4.

Specifically, the input of the learned regression model is the desired emotion strength and the output will be a set of facial landmark positions. To edit the 2D image, we first compute the difference between the facial landmark positions on the original input image and the output of the regression model. Then, the differences of the landmark positions will be used for warping the 2D image by Large Displacement Optical Flow (LDOF) [31]. Samples of edited images will be presented in Section 6.6.

6 Experimental Results

In this section, we present the results obtained using the proposed method. Firstly, we visualize the motions projected to a low-dimensional space when training our motion synthesis models in Section 6.2. Next, we evaluate the performance of the learned ranking functions that contain the relationship between low-level motion features and emotion strength numerically in Section 6.3. Thirdly, we present the computational cost for the training process in Section 6.4. Then, new motions are synthesized using the learned motion synthesis models in Section 6.5. We further demonstrate the generality of our proposed framework by editing 2D face images in Section 6.6.

6.1 Implementation Details

We used the Toolbox for Dimensionality Reduction library [32] to reduce the feature dimension from 198-d to 40-d. For the relative attribute ranking, we used the Matlab implementation provided by the authors of [9]. The proposed framework was implemented in Matlab R2015a [27] and all the experiments were conducted on a 64-bit machine with Intel Xeon 2.4GHz (E5-2620) and 64GB memory. The experiments ran on a single thread without any performance boost.

6.2 Visualization of the Low-dimensional Space

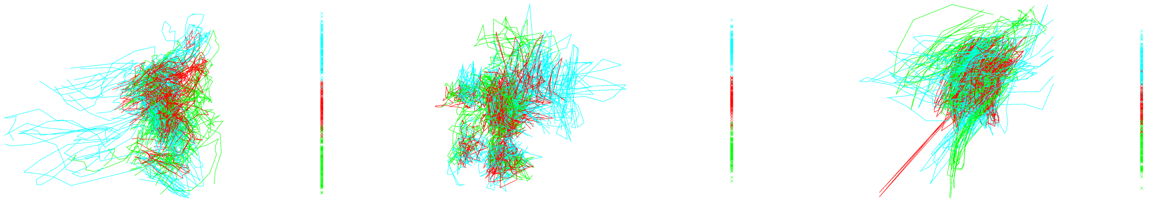
In order to learn the ranking function efficiently and effectively, the motion features are projected into latent space with a much lower dimensionality. In the experiments, each pose (frame) is represented by a 40-d in the latent space (198-d in the original motion). The dimensionality in the latent representation is selected empirically to balance the trade-off between training time and reconstruction error rate. We use 10 as the hidden layer size in the Neural-Network regression for training the motion synthesis model. We selected 10 empirically. We visually inspected the visual quality of the synthesized motions and the ranking accuracy quantitatively. We found that the performance does not improve with bigger hidden layer size and we selected 10 when implementing the framework. The latent representation of the motions with different emotions are illustrated in Figure 4 and 5 (left columns). Notice that only the first 3 principal components are displayed.

However, due to the style differences and variations between the subjects, the motions, which are represented as curves, with different emotion labels are mixed together. This shows that directly interpolating motions in the low dimensional space may result in significant change in the emotion strength as the motions (i.e. trajectories) with different emotions (i.e. colors) are tangled.

To facilitate motion synthesis by controlling the emotion strength, our proposed framework must be able to learn the important motions features that affect the emotion strength. The relationship can be learned by training a ranking function as explained in Section 4.3. The ranking function will be used for calculating the motion strength (a scalar value) of each motion. Then, the motions are ranked by the motion strength as shown in Figure 4 and 5 (right columns). The results indicate that the learned ranking function can effectively evaluate the emotion strength of each motion and is able to separate motions from different emotion strength. This facilitates the motion synthesis process as the ranking function contains the weights of each motion feature that reflects how much each feature contributes to the change in the emotion strength.

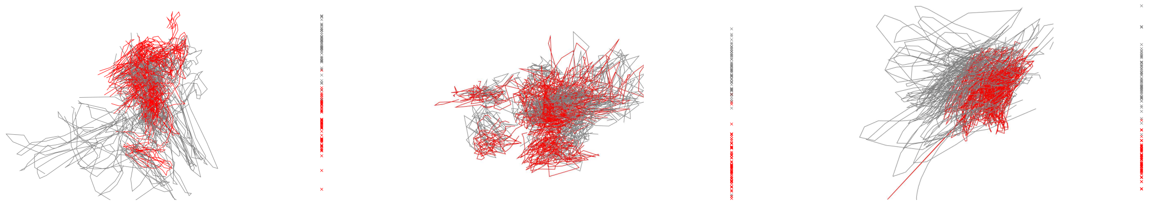
6.3 Evaluating the Learned Ranking Function

In addition to visualizing the effectiveness of the learned ranking functions on evaluating the emotion strength, we further evaluate the performance of the ranking function numerically. Specifically, we split the collected motions into 2 sets with equal numbers of motions - one set for training and the other set for testing such that the testing motions are 'unseen' data to the ranking function. Next, we use the labels of the training motions to setup the relative constraints as explained in Section 4.3. We vary the percentage of training data to train



(a) Throwing Motion (b) Knocking Motion (c) Lifting Motion

Figure 4: Visualizing the closeness of the training motions in (left column in each motion type) latent space (projected by PCA) and (right column) sequential order based on the emotion strength computed using the learned **happiness** ranking function on different motion types. Each keyframe is projected to the latent space using PCA and each of the polygon line in the figure is representing a motion sequence. Motions are colored according to the ground truth emotion labels: Happy (cyan), Neutral (red) and Sad (green).



(a) Knocking Motion (b) Lifting Motion (c) Throwing Motion

Figure 5: Visualizing the closeness of the training motions in (left column in each motion type) latent space (projected by PCA) and (right column) sequential order based on the emotion strength computed using the learned **anger** ranking function on different motion types. Motions are colored according to the ground truth emotion labels: Angry (grey) and Neutral (red).

the ranking function by randomly selecting the samples without considering the emotion strength. Then, the learned ranking function is used for computing the emotion strength of the testing motions. Finally, we compare the computed relative emotions strength to the labels of the testing motions and obtain the accuracy in ranking the testing motions. The same process is conducted on 3 different types of features, namely *PCA 80-d*, *PCA 40-d* and *no PCA*. The experiment is repeated 3 times and the average accuracy is reported. The results are shown in Table 1 and 2. They indicate that high ranking accuracy can be achieved in which the learned ranking function can be generalized to unseen data. This highlights the robustness of the ranking function. Moreover, the results also show that using PCA to reduce the dimensionality of the frame-based features does not have a significant impact on the ranking accuracy while improving the efficiency of the training process. We also vary the number of relative constraints used to train the ranking function as in [33] to reflect the model generality and the results show that there are no significant differences in the ranking accuracy. This further highlights the advantage of using our system as only a relatively small amount is needed to train the ranking function.

6.4 Computational Cost for the Training Tasks

In this section, we present the computational costs for the model training tasks. Table 3 shows the computation time required for training the ranking function explained in Section 4.3. The results show a significant reduction in training time when compared with the full

motion	frame	% of training pairs				
type	feature	10%	30%	50%	70%	90%
Knocking	PCA 40-d	71.3%	70.1%	69.5%	69.2%	68.7%
	PCA 80-d	71.9%	71.2%	70.8%	70.5%	70.4%
	no PCA	72.5%	72.0%	71.7%	71.5%	71.3%
Lifting	PCA 40-d	72.7%	73.1%	73.1%	73.1%	73.3%
	PCA 80-d	72.1%	72.9%	73.1%	73.1%	72.9%
	no PCA	72.3%	72.8%	72.8%	72.7%	72.8%
Throwing	PCA 40-d	75.2%	75.6%	76.0%	76.0%	76.1%
	PCA 80-d	75.3%	75.6%	75.9%	75.9%	76.1%
	no PCA	75.3%	75.7%	75.9%	75.9%	76.0%

Table 1: Ranking accuracy of the trained **anger** ranking function on unseen motion data.

motion	frame	% of training pairs				
type	feature	10%	30%	50%	70%	90%
Knocking	PCA 40-d	78.3%	76.6%	75.9%	75.2%	75.0%
	PCA 80-d	80.6%	79.3%	78.9%	78.5%	78.4%
	no PCA	81.1%	80.1%	80.0%	79.6%	79.7%
Lifting	PCA 40-d	79.9%	79.1%	78.6%	78.1%	77.7%
	PCA 80-d	80.9%	81.0%	81.0%	80.9%	80.8%
	no PCA	81.4%	81.3%	81.3%	81.3%	81.1%
Throwing	PCA 40-d	76.5%	76.0%	75.3%	74.9%	74.7%
	PCA 80-d	76.8%	76.7%	76.2%	76.0%	75.8%
	no PCA	76.8%	76.8%	76.3%	76.0%	75.8%

Table 2: Ranking accuracy of the trained **happiness** ranking function on unseen motion data.

motion type	emotion model	computation time (s)		
		PCA 40-d	PCA 80-d	no PCA
Knocking	Anger	3.09	15.09	152.41
	Happiness	5.13	15.27	106.17
Lifting	Anger	2.33	14.77	116.75
	Happiness	5.43	17.40	64.48
Throwing	Anger	2.70	15.02	108.00
	Happiness	5.05	17.91	70.08

Table 3: Computational cost (in seconds) for training the ranking function using different motion features.

feature setup (i.e. no PCA). This further highlights the performance gain using the proposed dimensionality reduction approach.

Table 4 shows the computation time required for training the non-linear regression function explained in Section 4.4. The training time varies from 10 to 15.5 minutes. Nevertheless, this training process is performed in the pre-processing stage and thus the training time is acceptable. A perform gain is expected when a parallel implementation of non-linear regression is used.

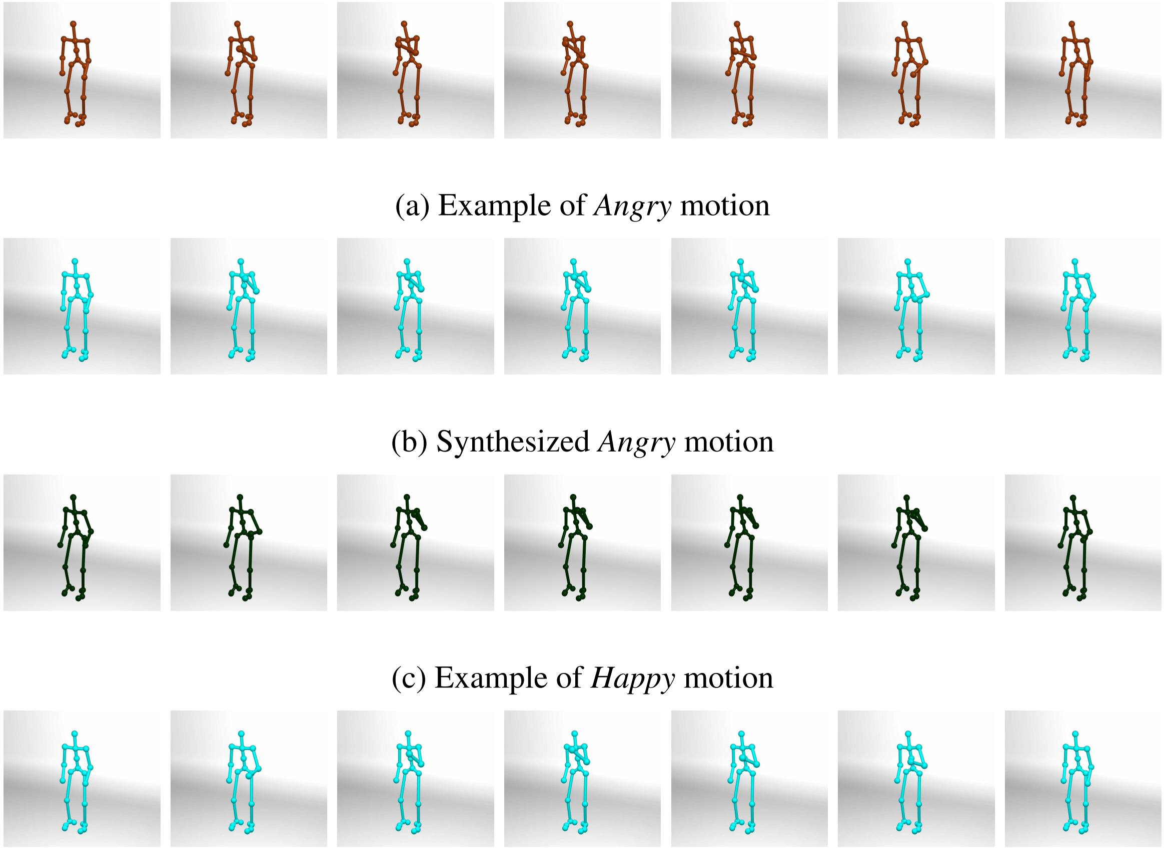
motion	emotion	computation time (s)
type	model	PCA 40-d
Knocking	Anger	491.01
	Happiness	601.83
Lifting	Anger	915.60
	Happiness	631.12
Throwing	Anger	889.16
	Happiness	647.27

Table 4: Computational cost (in seconds) for training the non-linear regression function using different motion features.

6.5 Synthesizing Motions with Different Emotion Strength

In this section, we show a number of motions with different emotion strength synthesized by our proposed framework. As explained in Section 4.4, a regression function is trained using the emotion strength and features obtained from the training data. At run-time, new motions can be created by specifying target emotion strength. We compared the synthesized motions with the training data at different emotion strength. Screen shots of training data and synthesized motions are shown in Figures 6, 7 and 8. The results show that our proposed method can synthesize motions with emotions that are comparable to the training data. More results can be found in the video demo accompanying with this paper.

We further visualize the trajectory of the left hand in all of the synthesized motions in



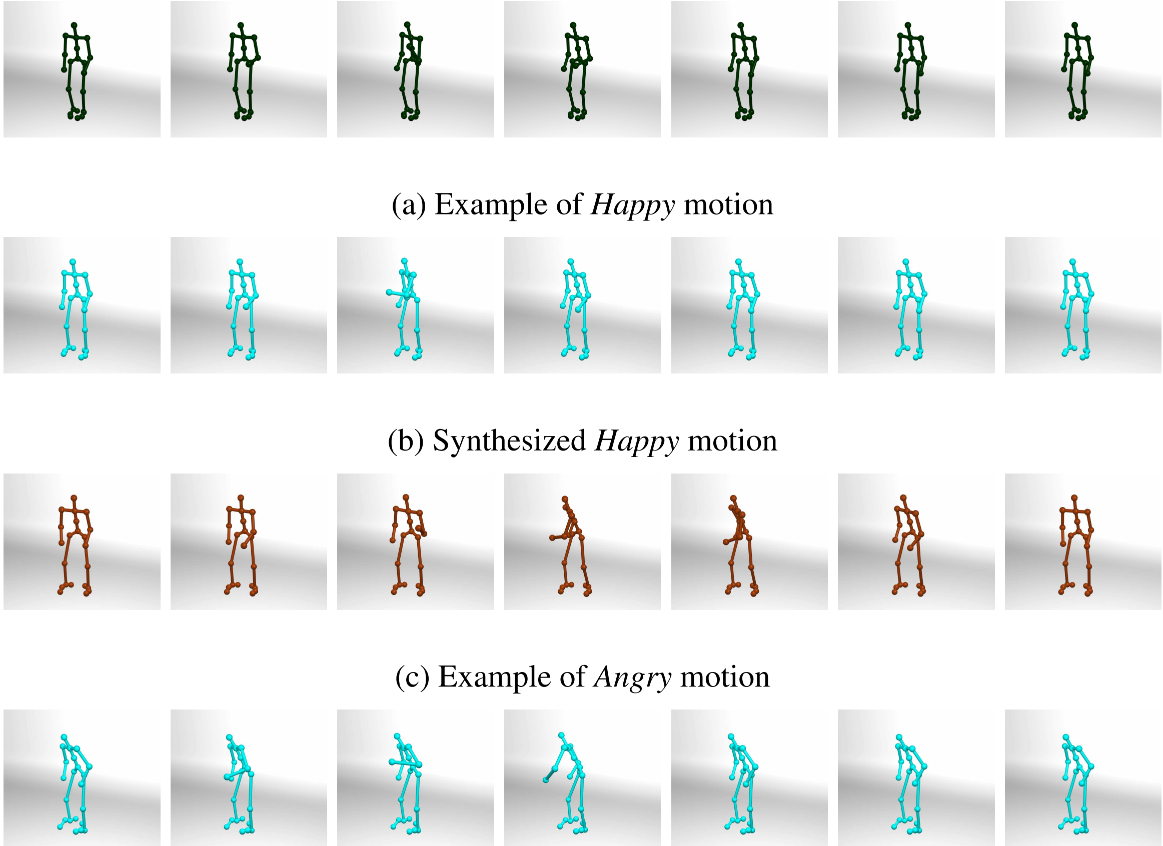
(a) Example of *Angry* motion

(b) Synthesized *Angry* motion

(c) Example of *Happy* motion

(d) Synthesized *Happy* motion

Figure 6: Knocking motions with (a) *Angry* and (c) *Happy* emotion expressions in the training data and the corresponding synthesized motions (b) and (d) created by our proposed method.



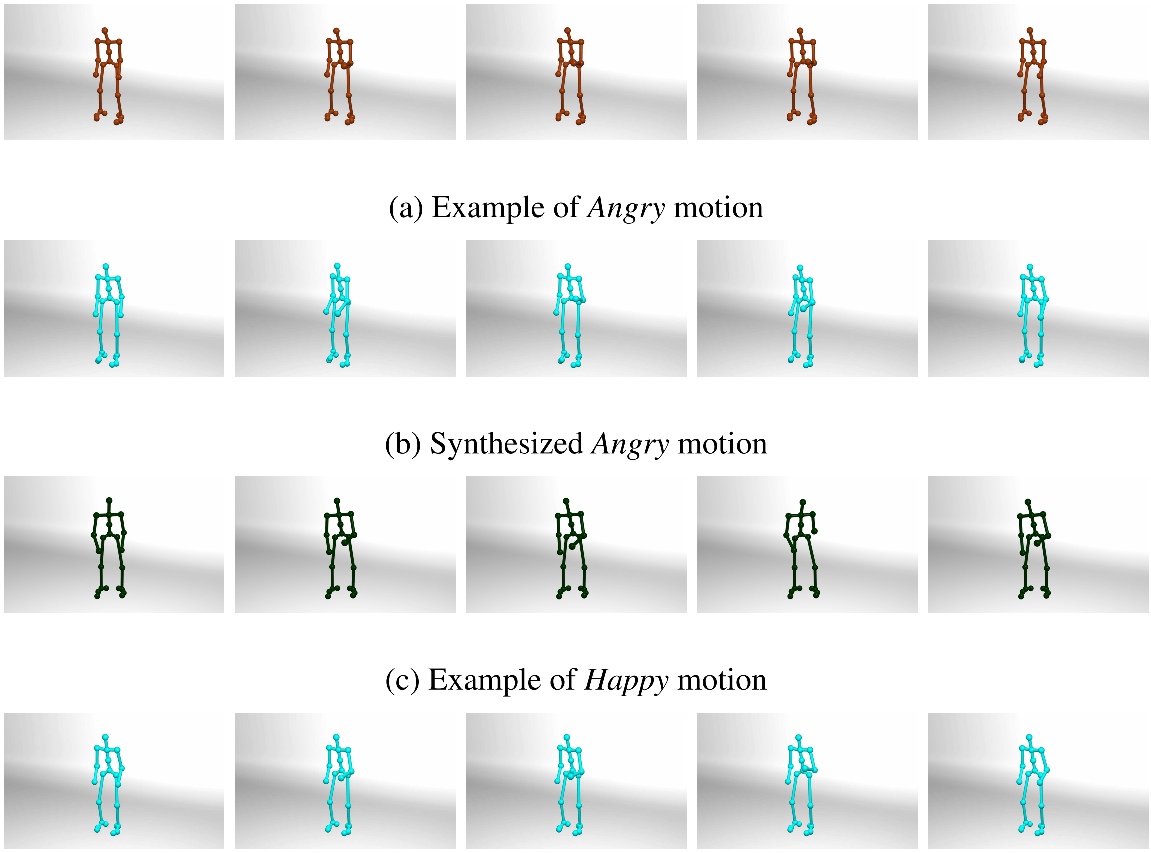
(a) Example of *Happy* motion

(b) Synthesized *Happy* motion

(c) Example of *Angry* motion

(d) Synthesized *Angry* motion

Figure 7: Throwing motions with (a) *Angry* and (c) *Happy* emotion expressions in the training data and the corresponding synthesized motions (b) and (d) created by our proposed method.



(a) Example of *Angry* motion

(b) Synthesized *Angry* motion

(c) Example of *Happy* motion

(d) Synthesized *Happy* motion

Figure 8: Lifting motions with (a) *Angry* and (c) *Happy* emotion expressions in the training data and the corresponding synthesized motions (b) and (d) created by our proposed method.

Figures 9-11 by showing a sphere on the 3D positions of the hand at every frame. The spheres are coloured according to the speed in that frame (red refers to a higher speed while blue refers to a lower speed). The spheres are the hand joint positions sampled evenly on the time domain, and the difference in the speed between different synthesized motions over time can further be highlighted by the sparsity of the spheres on the hand trajectory (the more sparse the spheres are, the faster the motion is).

Figure 9 illustrates the hand trajectories of throwing motions synthesized by adjusting the emotion strength. In Figure 9 (a), the hand trajectory becomes sparser and more spheres are colours in red when increasing the anger level (from left to right). It implies the hand is moving faster when the anger level increases. Also, the strongest anger level shows the motion exaggerated more. These observations align with our observations in real-life. The relationship between the speed of the hand and emotion strength also applies to happiness as shown in Figure 9 (b). However, we can see the difference in the hand trajectories in the motions synthesized by anger and happiness. All motions synthesized by adjusting the happiness level are having similar hand trajectories. This result highlights the effectiveness in synthesizing a wide variety of motion by adjusting the emotion strength using our proposed method.

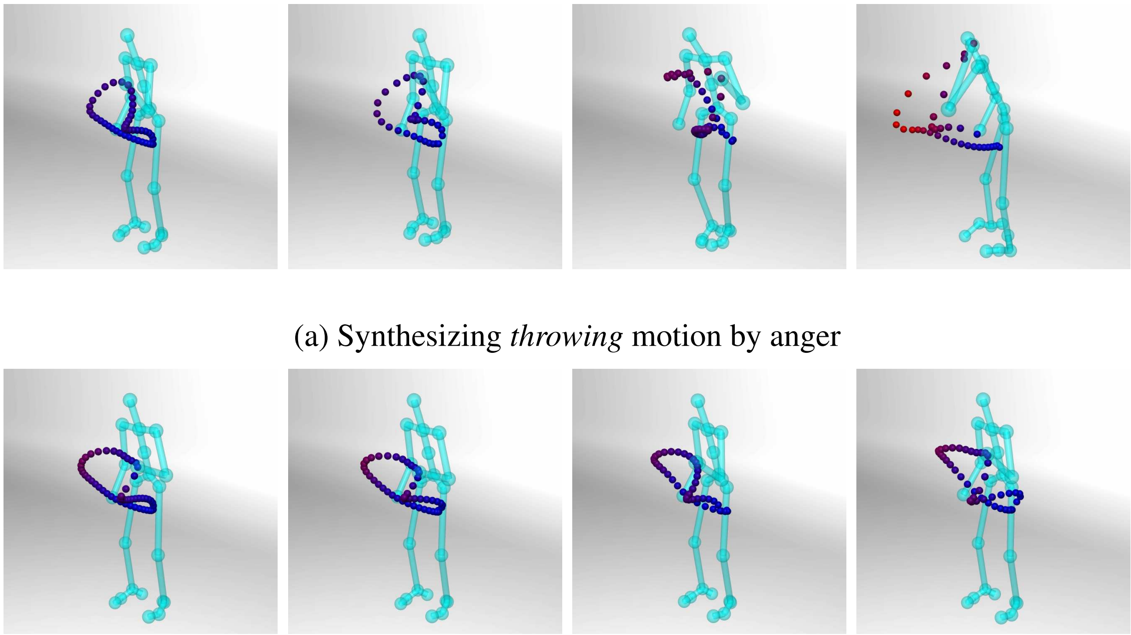
Figure 10 illustrates the hand trajectories of lifting motions synthesized by adjusting the emotion strength. In general, the lifting motions synthesized by adjusting the happiness level (Figure 10 (a)) are having a slower hand movement (i.e. denser sphere trajectories) than those synthesized by adjusting the anger level (Figure 10 (b)). The synthesized motions also showed a consistent trend in increasing the speed of the motion when the emotion strength increases.

Figure 11 illustrates the hand trajectories of knocking motions synthesized by adjusting the emotion strength. In general, the knocking motions synthesized by adjusting the happiness level (Figure 11 (a)) are having a more similar hand trajectory than those synthesized by adjusting the anger level (Figure 11 (b)). The knocking motions synthesized by adjusting the anger level showed an interesting pattern in which the character is a bit hesitant in the motions with lower anger level. The difference in the hand trajectories again highlights the effectiveness of our proposed method.

6.6 Editing 2D Images using Attribute Strength

In this section, we present the results on editing 2D face images using the proposed framework. We used the Labeled Faces in the Wild (LFW) [28] dataset in our experiment. The dataset contains 13233 images captured from 5749 subjects. We divided the dataset into two halves as training and testing sets by randomly assigning images to the image sets. The attributes of each image are calculated using the method proposed by Kumar et al. [29] and train the ranking function and regression model using the methods explained in Section 5.

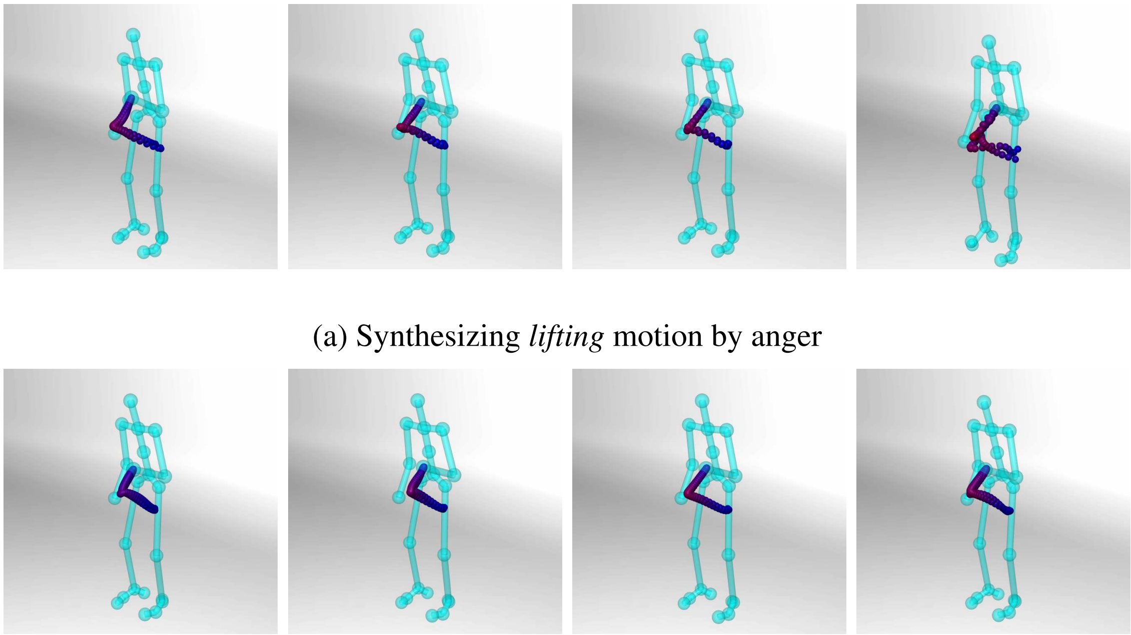
At run-time, given an input image and the desired change in attribute strength value, the facial landmarks are extracted using [30]. Next, the facial landmarks are converted into local coordinates by using the average positions of the facial landmarks and the local



(a) Synthesizing *throwing* motion by anger

(b) Synthesizing it throwing motion by happiness

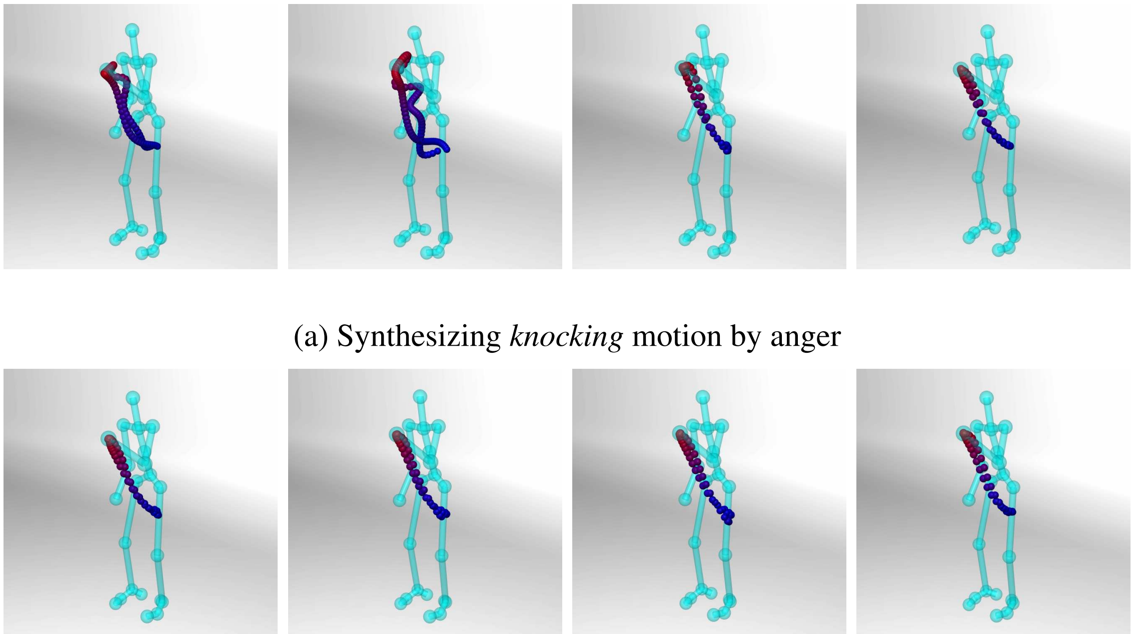
Figure 9: Synthesizing *throwing* motion by increasing emotion strength (from left to right) in (a) anger and (b) happiness. The new generated motions are best viewed and understood from the supplement videos.



(a) Synthesizing *lifting* motion by anger

(b) Synthesizing it lifting motion by happiness

Figure 10: Synthesizing *lifting* motion by increasing emotion strength (from left to right) in (a) anger and (b) happiness.



(a) Synthesizing *knocking* motion by anger

(b) Synthesizing it knocking motion by happiness

Figure 11: Synthesizing *knocking* motion by increasing emotion strength (from left to right) in (a) anger and (b) happiness.

coordinates are compared to those 12 centroids computed in Section 5.2. The model trained with the cluster which is closest to the input face image will be selected to alleviate the pose variations problem. Since the models are learned offline, the face image can be edited at interactive rate at run-time.

Examples of the edited images are shown in Figure 12 and 13. The results show that our proposed framework is general and able to edit images by simply specifying the desired change in attribute strength.

7 Conclusion and Discussions

In summary, this paper presents a new data-driven approach to learn the underlying relationship between low-level features and high-level emotion expressions at different level of strength. Our method takes the advantages of using relative attribute [9] to learn a ranking function that evaluates the attributes strength from low-level motion features. Since only weak relative constraints are used in the training process, the training data is not necessarily labeled with ground truth attribute strength as in the data set we used. Our method further makes use of the computed attribute strength for motion and facial expression synthesis. Once the model is learned in the pre-processing stage, synthesis can be done by specifying the target attribute strength at run-time. Our method can be applied to a wide variety of applications such as animation and movie production, as well as interactive applications such as computer games and virtual reality applications.

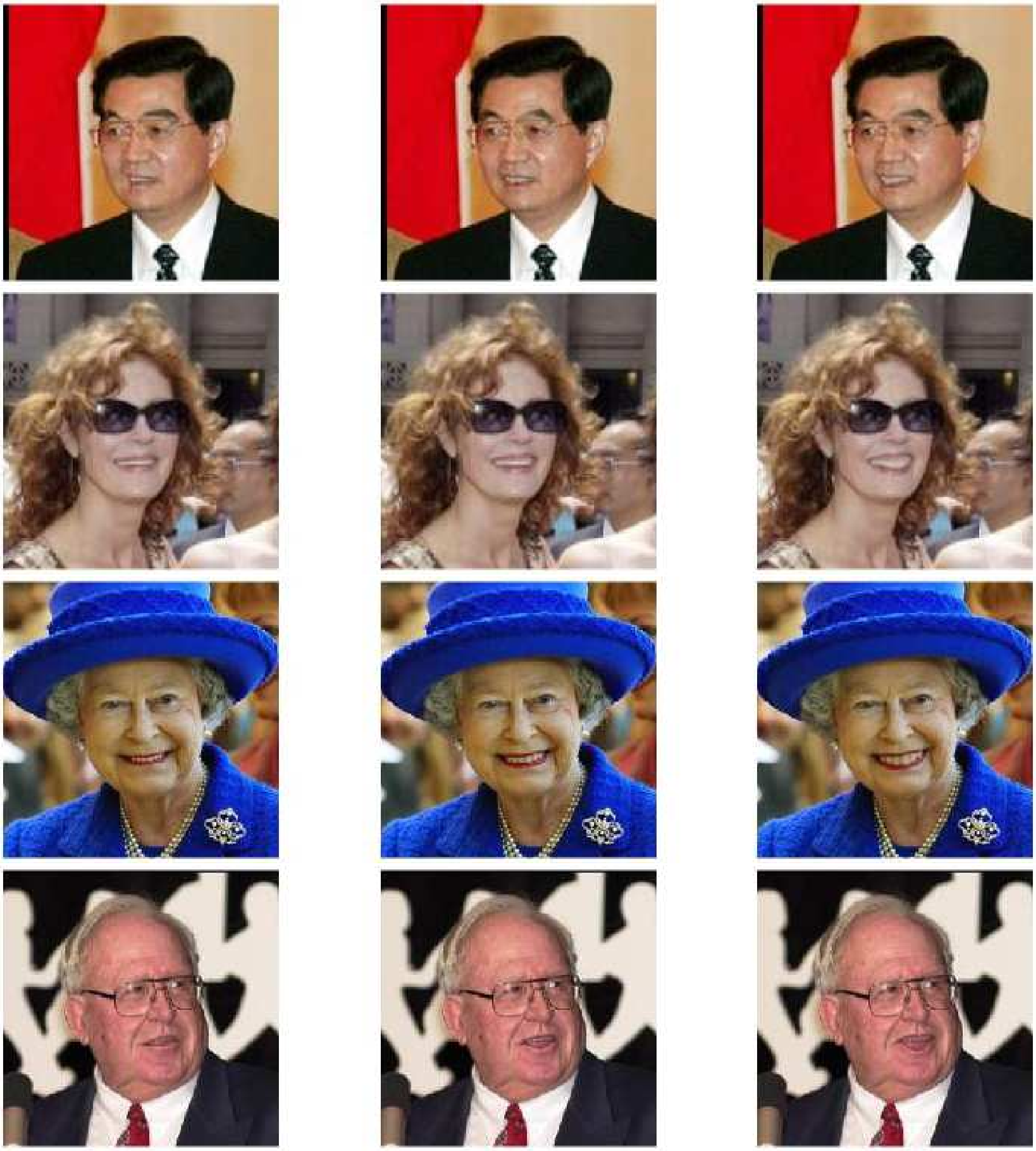


Figure 12: Examples of the edited 2D face images. Left column: less happy (edited). Middle column: input (original). Right column: happier (edited).

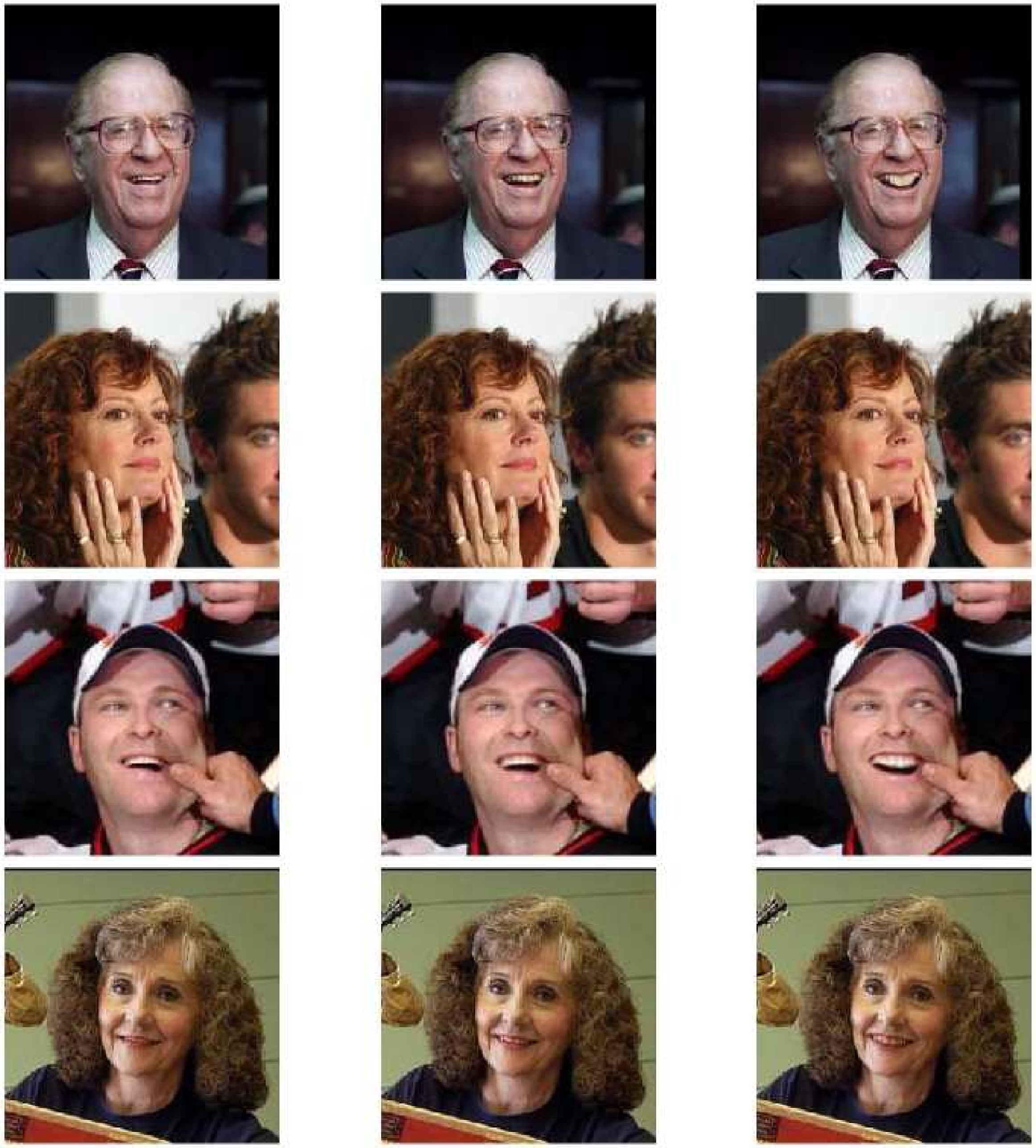


Figure 13: Examples of the edited 2D face images. Left column: less happy (edited). Middle column: input (original). Right column: happier (edited).

As the first attempt and a preliminary study on analyzing and utilizing learned attribute strength for motion and facial expression synthesis, we only focused on learning a single attribute from the data. However, as shown in previous studies in recognition tasks [33], taking into account the correlations between multiple attributes can improve the attribute learning performance. In our application, such an approach could enable the synthesis with multiple types of emotion expression, which is an interesting future research direction. Another future direction is conducting a user study to evaluate the emotion strength level of the synthesized results for validating the ranking function learned by our method. We are also planning to explore other possible low-level motion and facial features to be included in the learning process.

Acknowledgements

This work is supported in part by the Engineering and Physical Sciences Research Council (EPSRC) (Ref: EP/M002632/1) and the Royal Society (Ref: IE160609). We also thank Donald Chan for conducting the experiments.

References

[1] Markus Raab. Chapter 1 - the building blocks of performance: An overview. In Markus Raab, , Babett Lobinger, , Sven Hoffmann, , Alexandra Pizzera, , and Sylvain Laborde,

editors, *Performance Psychology*, pages 3 – 10. Academic Press, San Diego, 2016.

[2] Rachel McDonnell, Sophie Jörg, Joanna McHugh, Fiona Newell, and Carol O'Sullivan. Evaluating the emotional content of human motions on real and virtual characters. In *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*, APGV '08, pages 67–74, New York, NY, USA, 2008. ACM.

[3] Andreas Aristidou and Yiorgos Chrysanthou. Motion indexing of different emotional states using lma components. In *SIGGRAPH Asia 2013 Technical Briefs*, SA '13, pages 21:1–21:4, New York, NY, USA, 2013. ACM.

[4] Cathy Ennis, Ludovic Hoyet, Arjan Egges, and Rachel McDonnell. Emotion capture: Emotionally expressive characters for games. In *Proceedings of Motion on Games*, MIG '13, pages 31:53–31:60, New York, NY, USA, 2013. ACM.

[5] Aline Normoyle, Fannie Liu, Mubbasir Kapadia, Norman I. Badler, and Sophie Jörg. The effect of posture and dynamics on the perception of emotion. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '13, pages 91–98, New York, NY, USA, 2013. ACM.

[6] Shihong Xia, Congyi Wang, Jinxiang Chai, and Jessica Hodgins. Realtime style transfer for unlabeled heterogeneous human motion. *ACM Trans. Graph.*, 34(4):119:1–119:10, July 2015.

[7] M. Ersin Yumer and Niloy J. Mitra. Spectral style transfer for human motion between independent actions. *ACM Trans. Graph.*, 35(4):137:1–137:8, July 2016.

[8] Vittorio Ferrari and Andrew Zisserman. Learning visual attributes. In *Advances in Neural Information Processing Systems*, pages 433–440, 2008.

[9] D. Parikh and K. Grauman. Relative attributes. In *2011 International Conference on Computer Vision*, pages 503–510, Nov 2011.

[10] Matthew Brand and Aaron Hertzmann. Style machines. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 183–192, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[11] Raquel Urtasun, Pascal Glardon, Ronan Boulic, Daniel Thalmann, and Pascal Fua. Style-based motion synthesis. *Computer Graphics Forum*, 23(4):799–812, 2004.

[12] Eugene Hsu, Kari Pulli, and Jovan Popović. Style translation for human motion. *ACM Trans. Graph.*, 24(3):1082–1089, July 2005.

[13] Leslie Ikemoto, Okan Arikan, and David Forsyth. Generalizing motion edits with gaussian processes. *ACM Trans. Graph.*, 28(1):1:1–1:12, February 2009.

[14] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Trans. Graph.*, 33(4):149:1–149:11, July 2014.

[15] Leonid Sigal, Moshe Mahler, Spencer Diaz, Kyna McIntosh, Elizabeth Carter, Timothy Richards, and Jessica Hodgins. A perceptual control space for garment simulation. *ACM Trans. Graph.*, 34(4):117:1–117:10, July 2015.

[16] T. Kaneko, K. Hiramatsu, and K. Kashino. Generative attribute controller with conditional filtered generative adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7006–7015, 2017.

[17] Hui Ding, Kumar Sricharan, and Rama Chellappa. Exprgan: Facial expression editing with controllable expression intensity. *AAAI*, 2018.

[18] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F. Cohen. Bringing portraits to life. *ACM Trans. Graph.*, 36(6):196:1–196:13, November 2017.

[19] Tiziano Portenier, Qiyang Hu, Attila Szabó, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *ACM Trans. Graph.*, 37(4):99:1–99:13, July 2018.

[20] Yingliang Ma, Helena M. Paterson, and Frank E. Pollick. A motion capture library for the study of identity, gender, and emotion perception from biological motion. *Behavior Research Methods*, 38(1):134–141, 2006.

[21] Edmond S.L. Ho, Jacky C.P. Chan, Donald C.K. Chan, Hubert P.H. Shum, Yiu-ming Cheung, and Pong C. Yuen. Improving posture classification accuracy for depth

sensor-based human activity monitoring in smart environments. *Computer Vision and Image Understanding*, 148(C):97–110, July 2016.

[22] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Trans. Graph.*, 21(3):473–482, July 2002.

[23] Yiming Yang, Qiang Yang, Wei Lu, Jialin Pan, Rong Pan, Chenhui Lu, Lei Li, and Zhenxing Qin. Preprocessing time series data for classification with application to crm. In Shichao Zhang and Ray Jarvis, editors, *AI 2005: Advances in Artificial Intelligence*, pages 133–142, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[24] Ik Soo Lim and D. Thalmann. Key-posture extraction out of human motion data. In *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, volume 2, pages 1167–1169 vol.2, 2001.

[25] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph.*, 23(3):514–521, August 2004.

[26] Olivier Chapelle. Training a support vector machine in the primal. *Neural Comput.*, 19(5):1155–1178, May 2007.

[27] MATLAB. *version 8.5.0 (R2015a)*. The MathWorks Inc., Natick, Massachusetts, 2015.

[28] Gary B. Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, Marseille, France, October 2008. Erik Learned-Miller and Andras Ferencz and Frédéric Jurie.

[29] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372, Sept 2009.

[30] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2879–2886, June 2012.

[31] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, March 2011.

[32] L.J.P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review. *J Mach Learn Res*, 10:66–71, 2009.

[33] L. Chen, Q. Zhang, and B. Li. Predicting multiple attributes via relative multi-task learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1027–1034, June 2014.