Emotionally Expressive Motion Controller for Virtual Character Locomotion Animations

Diogo Silva
Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
diogo.goncalves.silva@tecnico.ulisboa.pt

Pedro A. Santos
INESC-ID & Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
pedro.santos@tecnico.ulisboa.pt

João Dias
Faculty of Science and Technology
University of Algarve
Faro, Portugal
jmdias@ualg.pt

***Abstract*—Style and emotional expressiveness are essential aspects of virtual character computer animation. For a virtual character to display different emotions, motion capture data conveying each desired style has to be recorded, even if the baseline motion is the same. Animators then have to refine and conjoin each recording in order to create the final animations making it a timely and costly process. Although there have been efforts made into the automatic generation of motions through Deep Reinforcement Learning techniques, the problem persists that, for each new desired emotion, reference data displaying said emotion has to be readily available and a new motion has to be learned from scratch. By combining Machine Learning with Emotion Analysis - in particular Laban Movement Analysis and the Pleasure, Arousal, Dominance Emotion State Model - we have developed a system that is capable of not only identifying the perceived emotion of virtual character locomotion animations but that also allows us to alter the character's expressed emotion in real time and without the need of additional data.**

***Index Terms*—computer animation, machine learning, kinematic models, physics-based models, sentiment analysis, motion synthesis**

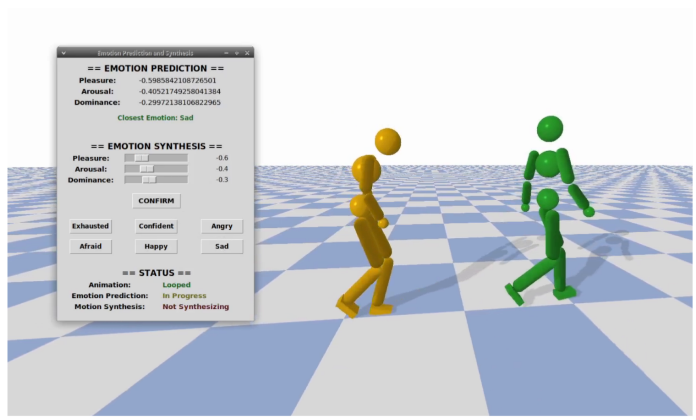

Fig. 1. The proposed system showcasing a reference baseline motion (right) and a physics-enabled policy-controleld character (left) whose movement has been altered to showcase the desired emotion "Sad".

I. Introduction

Conventionally, 3D computer character animation is created by professional human artists who manually tweak a given character's body in key frames and interpolate between them. This process is commonly aided by the usage of motion capture data (mocap). These consist in recordings of human actors done in a way that their motions can be directly applied to a virtual character. This data, when available, can be used as the basis for the animation and heavily aids the artist in speeding up the animation process.

Physics-based character animation generation has been growing in popularity due to its ability to synthesize realistic and natural-looking motions using only reference mocap files, without the need of manual animation work. Recent advancements made in Deep Reinforcement Learning (DRL) algorithms have allowed for the construction of such systems [9], [13], [15], able to successfully learn and reproduce physically accurate motor skills in a plethora of motions such as dances, locomotion and other such body gymnastics.

However, one problem both these systems and traditional mocap driven animation struggle with is having the character express different emotions using the same motion. Styles and emotions are an important aspect of generating realistic, believable virtual characters. Animators are usually tasked with not only creating the baseline movement for the animation, but also controlling the character's body language in order to convey different emotional states, feelings and styles. Such expressiveness is paramount to properly conveying a story, setting a scene's tone and making it so the virtual character has an actual impact on the viewer.

The problem then lies in the fact that, should animators want their character to convey different emotions, they would need to record actors portraying the same motion in all desired emotions. For example, if an animator wants a character to walk sadly, happily and angrily, they need to gather mocap data of the same walking animation but with the actor conveying sadness, happiness and anger. They then need to generate an entirely new animation for each emotion, either by training a motion learning system or through manual computer animation. This process has to be repeated each time the animator wishes for their character to express a new emotion - if they now want their character to convey the feeling of pride they need to once again get more mocap data and generate a whole new animation to add to the stack.

We propose a novel solution to this problem that combines the usage of Machine Learning (ML) models and Laban Movement Analysis (LMA) [7] for emotional classification and motion generation. Changes to the motion are applied in real time and get computed after a new desired emotion is specified. New poses are synthesized for the character at each frame, forcing the character to express the desired emotion, whilst still maintaining the baseline motion and movement.

The developed framework, shown in Figure 1, focuses on locomotive motions - walking, running and dashing - and allows users to edit the virtual character's expressed style and emotion in real-time, any number of times, without slowing down or stopping the animation and without the need for any additional mocap data or motion learning training. Moreover, our system works not only with Kinematic mocap data but also automatically generated Physics-Enabled Policy based character controllers learnt using the Spacetime Bounds DRL system [13].

II. Related Work

A. Motion Learning

There have been numerous efforts poured into creating virtual character controllers that can automatically learn how to mimic and perform animations without the need of human animators. Earlier approaches focused on purely data-driven Kinematic Models generated by neural networks [11], [20]. More robust solutions based around Physics-Based models [2], [6], [15], [24] offered the guarantee of generating physically accurate motions. The state of the art now lies in the usage of Reinforcement Learning methodologies for the generation of physics-based character controllers. Systems such as ***DeepMimic*** [15] proved the efficacy of such techniques in creating policy-based character controllers able to imitate motions, provided via motion capture data. ***SpacetimeBounds*** [13] further iterated on the ideas of DeepMimic through the introduction of loose space-time constraint used to limit the training search space in a fashion akin to early termination. These restrictions bind the character's states in space and time during the reinforcement learning training process based only on the given reference motion. Additionally, by loosening or tightening the spacetime bounds, this system allows users to indirectly curate the look and feel of the outcome motion, hence providing a manner of style exploration.

An issue with SpacetimeBound's stylistic exploration is that after the character controller policy has been learned, there is no way to further edit the character's style or emotion. This issue is prevalent in all of the aforementioned systems which focused on learning to mimic the given references rather than empowering the character with the capability of expressing the same motion in a wide array of emotions. Our work aims to fix this issue by allowing users to edit and swap the learned animation's expressed emotion in real time, without the need of additional references or further training.

B. Motion Analysis and Tweaking

Emotional classification involves manners of distinguishing emotions from one another. There are two main approaches to emotion classification - one in which emotions are considered discrete, meaning humans have a preset array of emotions that they discretely swap between [10], [21], and one in which emotions are defined in accordance to continuous values in dimensional axis, blending into each other smoothly [14], [19]. Focusing on the latter, there are several dimensional models that attempt to place emotions on a 2D or 3D scale. Russel's Circumplex Model (RCM) [19] is one such model which maps emotions into a 2 dimensional space consisting of an Arousal and Valence axis, describing emotions alongside a Deactivated/Alert and a Pleasure/Displeasure continuum. The ***Pleasure, Arousal, Dominance Emotional State Model*** (PAD) [14] is an extension of the ideas of RCM, adding a new emotional dimension - Dominance. This new axis allows for a more granular specification of the character's emotion, accounting for the emotional impact of external forces upon the actor's feelings.

Motion analysis focuses on parameterizing and describing a character's movements. ***Laban Movement Analysis*** (LMA) is one such motion analysis methodology capable of describing human movements by drawing inspiration from fields of anatomy, kinematics and psychology [3], [7]. LMA breaks down movement description into 4 categories - Body, Effort, Shape and Space - each possessing different properties. Recent efforts have been able to utilize LMA features to accurately assess the discrete emotion of different gaits by further splitting the LMA features into Posture, and Movement features [17].

A noteworthy approach to motion analysis and tweaking is the one proposed by Aristidou et al. [3]. These authors developed a system capable of extracting a motion capture's select set of LMA features and mapped them into the RCM emotional coordinates through Linear Regression. They also managed to achieve the inverse process of mapping 2D emotional coordinates back into a set of LMA features. These generated LMA features were then fed into a Heuristic-Rules based motion synthesis algorithm, transforming them into joint rotation changes that could then be applied to the virtual character using Inverse Kinematics. Our system draws a lot of inspiration from this one but presents several key changes. Firstly, instead of focusing on contemporary dance motions our efforts diverged towards locomotion animations. Rather than using the RCM model we utilized the more descriptive PAD model [14]. Our selected set of LMA features also differs, taking notes from the efforts of Randhavane et al. [17] of gait emotional identification using LMA. Furthermore we utilized Gradient Tree Boosting Regression for LMA to PAD mapping and proposed the usage of both Gradient Tree Boosting Regressors and an Autoencoder to reduce the dimensionality of LMA features when mapping from PAD to LMA. Finally, our system allows for the emotional identification and tweaking of not only mocap driven kinematic controllers, but also policy-controlled physics-based characters.

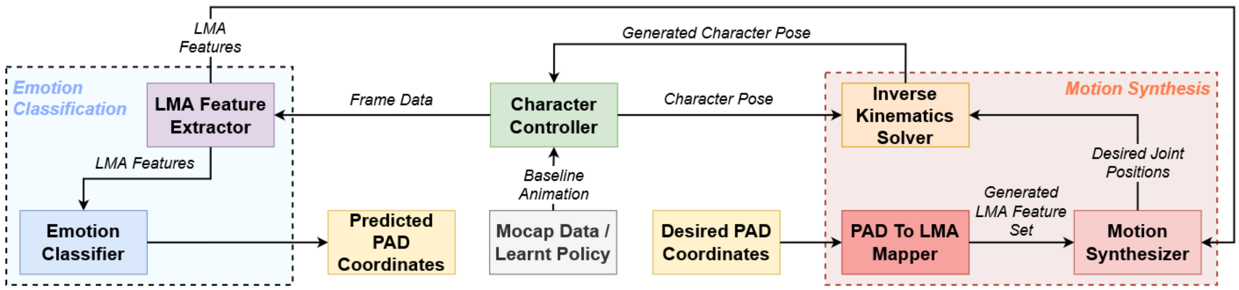

Fig. 2. An overview of the system's architecture.

III. Emotionally Expressive Motion Controller

The developed system can be subdivided into several core sub-modules. Figure 2 illustrates the connections between the modules and the system's overall architecture. At the core of the system lies a character controller used to make a character execute the intended baseline animation. This controller can either be Kinematic, driven directly by provided mocap, or Policy-based Physics-Enabled, learned using Spacetime Bounds [13]. For Emotion Classification, the system begins by computing a set of LMA features from the frame data extracted from the character. These features are then given to the Emotion Classifier module which outputs the predicted PAD coordinates. Emotional Motion Synthesis is triggered whenever new desired PAD coordinates are specified. Firstly the system converts the new coordinates into a set of LMA features. These features, alongside all of the baseline animation's LMA features, are then given to the Motion Synthesis module which computes new desired joint positions. The joint positions are then provided to the Inverse Kinematics Solver which uses them, alongside the character's current pose, to generate a new pose.

Following is a brief introduction to the system's main modules.

A. LMA Feature Extraction

This module receives Frame Data in the form of joint positions and transform them into a set of LMA Features.

B. Emotion Classifier

The Emotion Classifier module is equipped with our pre-trained LMA to PAD regression models. This module receives the current motion's LMA Features, standardizes them and feeds them to the LMA-PAD models. The resulting output is the predicted emotional PAD coordinates for the current LMA feature set.

C. PAD To LMA Mapper

This module is equipped with trained PAD to LMA machine learning models and is responsible for converting the input PAD coordinates into a set of corresponding LMA feature values.

D. Motion Synthesis

The Motion Synthesis module receives both the character's current and the generated LMA feature sets. Using these it then computes new desired core joints positions using a set of heuristic rules.

E. Inverse Kinematics Solver

This module takes the generated core joints positions alongside the character's current pose and uses Inverse Kinematics to compute a new pose. This pose aims to place the character's joints as close as possible to their desired positions.

IV. Implementation

A. Dataset

We utilized the **Bandai-Namco-Research-Motion Dataset** [4]. This data consists of Bounding Volume Hierarchy (BVH) files describing a wide array of motions such as walking, running, kicks and dances. Each animation was performed in order to convey a specific style such as active, masculine or proud. Only the Walking, Running and Dashing animations were utilized and converted from BVH into a Deepmimic friendly-format [12]. Each style was also mapped into a specific emotion with corresponding PAD emotional coordinates [8]. A total of **468 different animations** in **10 emotional styles** - Neutral, Tired, Exhausted, Angry, Happy, Sad, Proud, Confident, Afraid and Active - were obtained, each running at 30 frames per second. Figure 3 showcases a sample of 4 motions from our dataset displaying the Afraid, Happy, Sad and Proud emotions, alongside their corresponding Pleasure, Arousal and Dominance coordinate values.

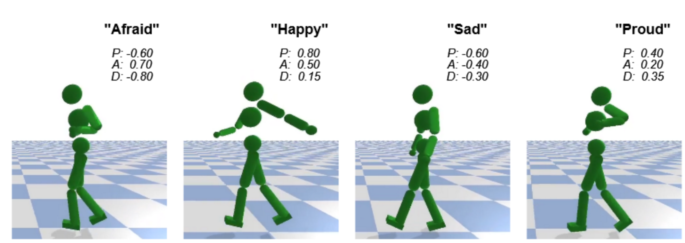

Fig. 3. Example of 4 motions from our dataset expressing 4 different emotions.

TABLE I
OUR SET OF 25 LMA FEATURES

LMA Feature	f	**LMA Category**
Max Hand Distance	f_1	Body
Avg. Left Hand - Hip Distance	f_2	Body
Avg. Right Hand - Hip Distance	f_3	Body
Max Stride Length	f_4	Body
Avg. Left Hand - Chest Distance	f_5	Body
Avg. Right Hand - Chest Distance	f_6	Body
Avg. Left Elbow - Hip Distance	f_7	Body
Avg. Right Elbow - Hip Distance	f_8	Body
Avg. Chest - Pelvis Distance	f_9	Body
Avg. Neck - Chest Distance	f_{10}	Body
Avg. Total Body Volume	f_{11}	Shape
Avg. Lower Body Volume	f_{12}	Shape
Avg. Upper Body Volume	f_{13}	Shape
Avg. Area between Hands and Neck	f_{14}	Shape
Avg. Area between Feet and Hip	f_{15}	Shape
Left Hand Speed	f_{16}	Effort
Right Hand Speed	f_{17}	Effort
Left Foot Speed	f_{18}	Effort
Right Foot Speed	f_{19}	Effort
Neck Speed	f_{20}	Effort
Left Hand Acceleration Magnitude	f_{21}	Effort
Right Hand Acceleration Magnitude	f_{22}	Effort
Left Foot Acceleration Magnitude	f_{23}	Effort
Right Foot Acceleration Magnitude	f_{24}	Effort
Neck Acceleration Magnitude	f_{25}	Effort

Upon labeling each of our animation files their LMA Features were extracted. First, each frame's pose information (joint positions, velocities and rotations) gets stored. At each keyframe - every 5th frame - the stored data is then used to compute the LMA features corresponding to these past frames. Each set is composed of **25 different LMA features** as specified in Table I. A total of **78551 LMA Feature Sets** were retrieved, each labeled according to their PAD coordinates.

B. Emotional Classification

To classify the motion's perceived emotion a set of Gradient Tree Boosting Regressors were trained to map LMA Features into PAD coordinates. We used 3 regressors - one for each emotional coordinate - that took as input our set of 25 LMA Features and outputting the corresponding predicted coordinate. Figure 4 illustrates this process. Regression was chosen over classification due to the fact that the PAD Model identifies emotions according to their Pleasure, Arousal and Dominance values which are continuous.

The models were built using XGBoost [5]. Our dataset of LMA Features was first standardized and then shuffled and split into a train and test set. 80% of data was used for Training (62841 samples) and 20% was used for Testing (15710 samples). Hyper parameter tuning was done individually for each of the 3 regressors using Random Search 10-Fold Cross Validation. The final models managed to accomplish a mean absolute error of 0.02, 0.06 and 0.03 using the Test set for the Pleasure, Arousal and Dominance coordinates correspondingly, with values ranging between $[-1.0, 1.0]$. The predicted emotional coordinates of 1000 random samples from

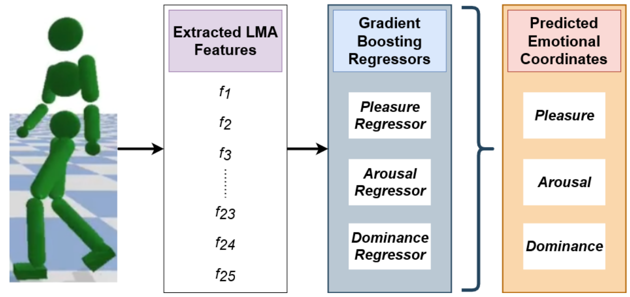

Fig. 4. The process of using our Gradient Boosting Regressors to predict the PAD coordinates of a set of LMA features extracted from a motion

our Test set are shown in Figure 5. As can be seen, some predictions do stray slightly from their real coordinates, but they nevertheless fall into well defined emotion clusters and seldom stray from the correct octant in the 3D model.

Using the trained predictors it is then possible to identify a given motion's perceived emotion in real time. During an animation's playtime LMA Features are extracted at every keyframe. After a list of 10 LMA Feature sets has been stored a new multithreaded process is started. This process standardizes the features and uses the predictors to compute the Pleasure, Arousal and Dominance coordinates for each of the sets. Each coordinate's predictions is then averaged, stored and output. At the end of the animation the final emotional prediction gets computed using a weighted average of all past predictions. For this average, the highest absolute recorded value for each emotional dimension is given a slightly higher weight. This stems from the assumption that the intensity of the intended emotional expression can vary throughout the course of the animation, but it will at some point reach a maximum absolute value, indicative of the feeling the character is aiming to express.

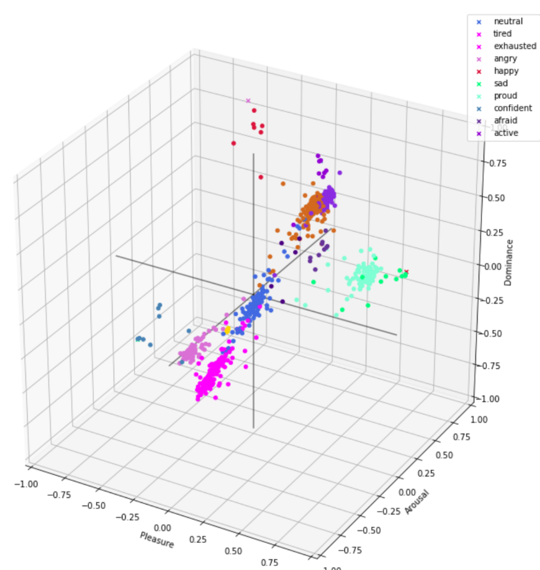

Fig. 5. Prediction results of samples from our Test set. Each sample is coloured according to their real emotion.

C. LMA Feature Generation

After training the LMA to PAD models, they were then used to generate a new dataset. This dataset stored our sets of LMA Features as target variables and their predicted PAD coordinates as inputs. New models capable of synthesizing new LMA Feature values from given PAD coordinates were trained using this data. First, an *Autoencoder* was created to convert the 25 LMA Features into a 5 dimensional Latent Feature space - l_1, l_2, l_3, l_4, l_5 - and vice-versa. This was done to decrease the overall complexity of the PAD-LMA mapping problem [22], [23]. A set of 5 Gradient Tree Boosting regressors was then trained to map PAD coordinates into each of these Latent Features. Figure 6 shows the process of generating new LMA values. The PAD coordinates are converted into Latent Features which in turn get decoded by the AutoEncoder into a set of corresponding LMA Features.

The Autoencoder Neural Network was built with the architecture illustrated in Figure 7. After training for 1024 epochs, it accomplished a mean absolute reconstruction error of 0.17 on the test set. We then generated a new labeled dataset using our PAD coordinates as input and the latent features created by the Autoencoder as output. This new dataset was used to train 5 regressors built using XGBoost in a manner similar to the predictors for Emotional Classification. Through this process we achieved an overall mean absolute error of 0.19 between the predicted emotional coordinates of the generated LMA Feature set and the original ones, with Pleasure an error of 0.19, Arousal 0.24 and Dominance 0.14.

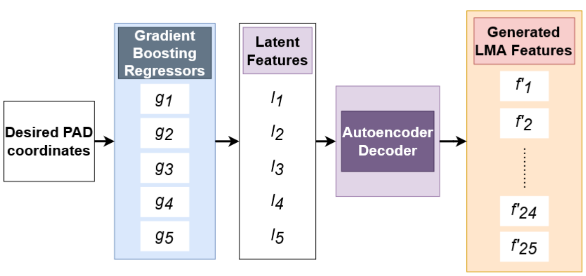

Fig. 6. Generation of LMA Features from PAD coordinates.

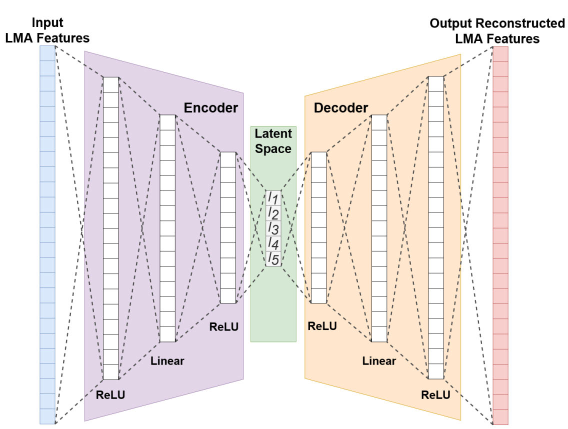

Fig. 7. The Autoencoder architecture.

D. Motion Synthesis

Given a new set of desired PAD coordinates it is possible to synthesize and apply motion changes to the character in real time. This editing can be performed multiple times and is done in a multithreaded process so as to avoid interrupting or slowing down the current animation's display. We designed a set of 6 Heuristic Rules, each responsible for tweaking the position or rotation of one of our core joints - Hips, Chest, Hands, Elbows, Feet and Neck. Changing upper body joints was the main focus as these tend to have the most impact on the conveyed emotion, while lower body joints are more important for balance and motion integrity rather than expression [3]. A subset of our rules can be seen in Figure 8. Each of these rules works by taking into account the current position or rotation of the joint its trying to change and one or more associated coefficients.

Whenever a new set of PAD coordinates is provided, new values for our set of LMA Features are created. These generated LMA Features. together with the animation's recorded LMA features, are utilized to compute the coefficients used in our heuristic rules. Each rule is associated with a different subset of LMA Features and its associated coefficients are computed by finding the value that minimizes the distance between the corresponding subset of recorded and generated LMA features. For example, rule g_1 aims to modify the position of our hips joint. To compute c_1, the coefficient associated with rule g_1, we find the value that minimizes the difference between the values of all recorded and generated LMA features that pertain to the hips. Looking at Figure I, for coefficient c_1, these features include $f_9, f_{11}, f_{12}, f_{13}$ and f_{15}. All coefficients are initialized at 1.0 and are minimized using Powell's method [16].

After computing the coefficients for each rule, the system then synthesizes the changes to the pose necessary to convey the desired emotion. If the character is being controlled by a learnt policy that means all poses are newly created at each frame. As such to get each frame's baseline pose we wait for it to be generated and interject it just before it gets applied to the character.

Rule	Affected LMA Features
$g_1(c_1)$: Modifies the hips height $r'_x = r_x$ $r'_y = r_y + (c_1 - 1.0) * 0.08$ $r'_z = r_z$ Where **r** is the current pelvis position and **r'** is the new desired pelvis position	$\mathbf{c_1}$: $f_9, f_{11}, f_{12}, f_{15}$

$g_2(c_2)$: Modifies the chest position
if $c_2 > 1.0$:
$w = 0.025$
else:
$w = 0.1$
$n'_x = n_x - (c_2 - 1.0) * w$
$n'_y = n_y + (c_2 - 1.0) * w$
$n'_z = n_z$
Where **n** is the current chest position and **n'** is the new desired chest position
	$\mathbf{c_2}$: f_9, f_{11}, f_{13}

Fig. 8. 2 of our 6 Motion Synthesis rules.

The heuristic rules are given the currently extracted pose to generate new core joint positions. These positions get handed to the Inverse Kinematics module to compute a new pose that attempts to get the core joints as close as possible to their desired synthesized positions, while still respecting the character's body restraints to avoid unnatural postures. The generated pose is then applied to the character, replacing the baseline pose and thus altering the character's emotional expression.

V. Results

Our system was built in Python 3.8, using PyBullet [1] as the underlying engine. All machine learning models were trained offline in a dedicated external server. Emotional Classification and Motion Synthesis is done in multithreaded processes and takes, on average, less than 3 seconds to execute and apply, running in real time. The system's test results, project code and other resources, were made publicly available[1].

To illustrate the functioning of our Emotionally Expressive Motion Controller system the Graphical User Interface (GUI) shown in Figure 9 was developed. Users are shown a window with the virtual character performing the specified motion. They can pan the camera around and zoom in and out. If the user specified a policy-based character controller an additional character is also placed alongside the main one, showcasing the reference motion the policy learned from. The GUI shows the current results of the Emotional Classification by displaying the predicted Pleasure, Arousal and Dominance coordinates. To specify new desired PAD coordinates the user can freely tweak the corresponding sliders or select one of the available presets. Hitting the confirm button triggers the Motion Synthesis with the coordinates currently on the sliders. Aside from this, system state information is also showcased. Specifically, the GUI indicates whether the animation is running, has looped or has stopped, whether Emotional Classification is still ongoing or has finished and whether a new motion is being synthesized or not.

Triggering the Motion Synthesis module will alter the character's motion in real time. Figure 10 showcases 4 generated motions synthesized from the same baseline animation. The "Confident" character, for example, highly elevates their shoulders, widens their upper body volume and exposes their neck, while the "Afraid" character raises their arms to protect their torso and slumps down, reducing its body volume. Our synthesis works best when applied to a neutral baseline movement but it nevertheless works with different base emotions. Both our emotional classification and our synthesis were trained and designed for locomotion-type motions, specifically walking and running. Whilst they can still be applied to other types of animations without additional changes, the results won't be as consistent. Motion Synthesis seems to suffer the most from this in the quality of the generated motions, mostly due to the fact that our heuristic rules were purposefully tweaked for

[1] https://heroufenix.github.io/expressive_animations_web/

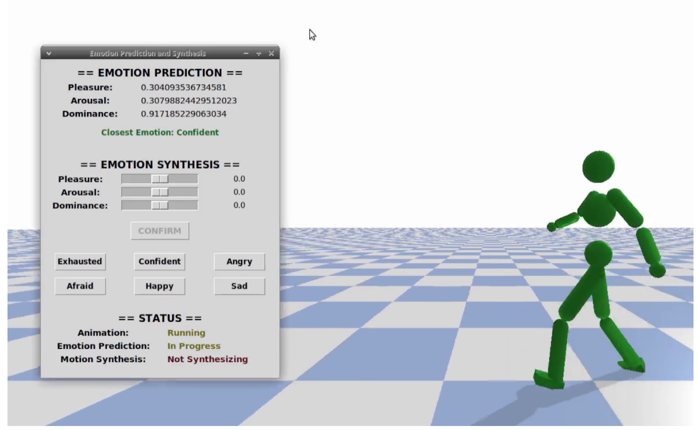

Fig. 9. Our system displaying motion capture data and our Graphical User Interface showing the current Emotional Prediction, system state and Motion Synthesis controls.

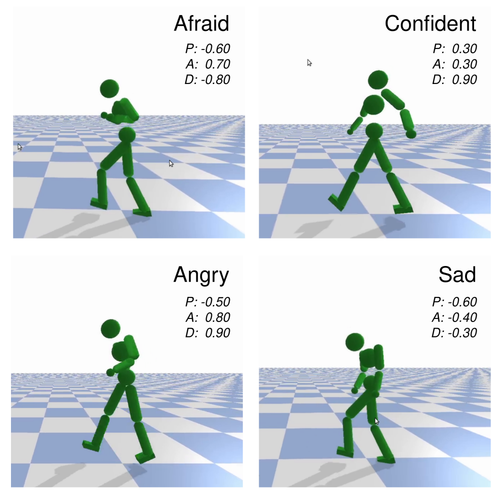

Fig. 10. Four motions synthesized using the same baseline motion and 4 different desired emotions.

locomotions. The Emotional Classification also suffers in the accuracy of its predictions but still manages to, more often than not, predict the correct emotional octant.

A. User Tests

The performance of our Emotionally Expressive Motion Synthesis was evaluated through a set of user tests. These tests aimed to compare the emotional expressiveness of the generated motions versus the ones from the mocap dataset, which were performed by professional actors to convey a specific emotional style. A set of video clips was recorded using our motion generation over both a Kinematic and a Policy-Controlled physics enabled character. The generated motions were created by altering the emotion of a baseline walking animation from Neutral into Sad, Confident, Tired, Afraid, Angry and Happy. The intent was to check whether the generated motions managed to convey their intended emotions as well as the reference mocap. Two tests were conducted with 40 anonymous participants each. For each test we utilized a form containing the recorded clips mixed together in a random order.

For the first test, participants were asked to visualize each clip and select which emotion they thought the character was trying to express from a given list. The results were gathered in the clustered bar charts shown in Figure 11. Looking at the reference mocap, most participants managed to correctly identify the emotions “Afraid”, “Confident”, “Happy” and “Tired”, although not by a vast majority in most cases. Compared to mocap, the generated motions applied to a Kinematic character managed to output better results, with the correct emotions being the most selected in all cases but “Tired” which nearly tied with “Sad”. As for the generated motions applied to a Policy-based controller, the best performing emotions were “Confident”, “Sad” and “Tired” with the remaining ones being in second or third place. In general, both generated models had similar performances comparatively to each other and to the reference mocap data.

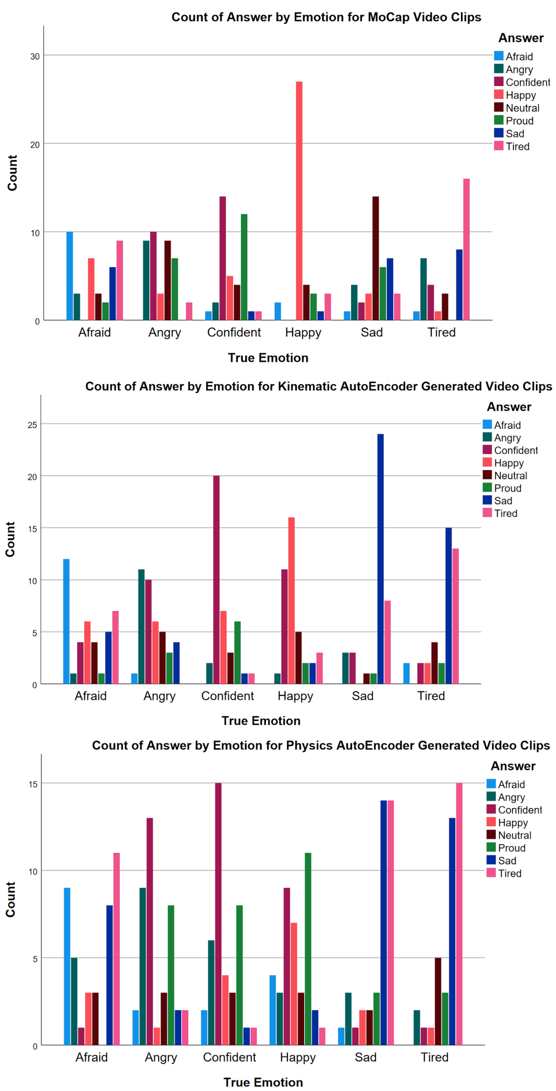

Fig. 11. Clustered bar charts showing the count of answers compared to the correct emotion for each type of clip.

Certain emotions have intrinsic ambiguity when lacking context which might explain some of the results obtained on the first test [18]. To counteract this, a second test was conducted where participants were explicitly told which emotion the character was trying to express. They were then asked to rate how much they agreed the character was expressing said emotion using a Likert scale from 1 (Completely Disagree) to 5 (Completely Agree). A Friedman test done for each emotion showed that there was no statistically significant difference between our video clip types for “Afraid” ($p = 0.519$), “Confident” ($p = 0.121$) and “Angry” ($p = 0.657$). An additional Wilcoxon Signed Rank Test on the remaining emotions - “Sad”, “Tired”, “Happy” - confirmed, for all cases, a statistically significant difference between the generated motions and the reference mocap ($p < 0.001$). Figure 12 exemplifies the dispersion of answers per type of clip for the emotions where a statistically significant difference between the type of clip was found. For the Sad and Tired emotions, both types of generation actually outperform the reference mocap meaning that for these particular emotions, our generated motions are more easily identified as their corresponding emotions. The “Happy” emotion was the only one in which mocap outperformed our generated motions, although they still had decent results with most participants agreeing that the character was in fact exhibiting “Happiness”.

Looking at the results of both tests, participants, for the most part, managed to correctly identify and tended to agree with the emotions that the generated motions were trying to convey. Moreover, certain emotions were more easily identified comparatively to the reference mocap. This showcases the efficacy of our system, as it proves that we can achieve results with similar emotionally expressiveness to professional-grade mocap without the need and costs of recording several actors performing each of the desired emotions.

VI. Conclusion

We have showcased our system for Emotional Classification and Emotionally Expressive Motion Control of Locomotion animations. We have proven that, through the usage of select LMA features we can accurately identify a character’s expressed emotion in the 3D PAD Emotional space. We also managed to create a methodology for generating a new set of LMA Features with desired emotional values and use it to alter a character’s motion in real time and without requiring any additional data or training. Furthermore, our system works not only on Kinematic controllers driven by mocap, but also on physics-enabled characters controlled by learnt policies.

Our system’s value lies in the fact that we can alter a motion’s emotion in real time without the need for any further data or training. Our system bypasses the need of having to record a mocap or train a character controller policy for each emotion that the character is meant to express over the same motion by managing to change the character’s emotion instantaneously while its still performing the baseline movement. To showcase our emotional classification and expressive motion editing we also designed an easy to use and interactable

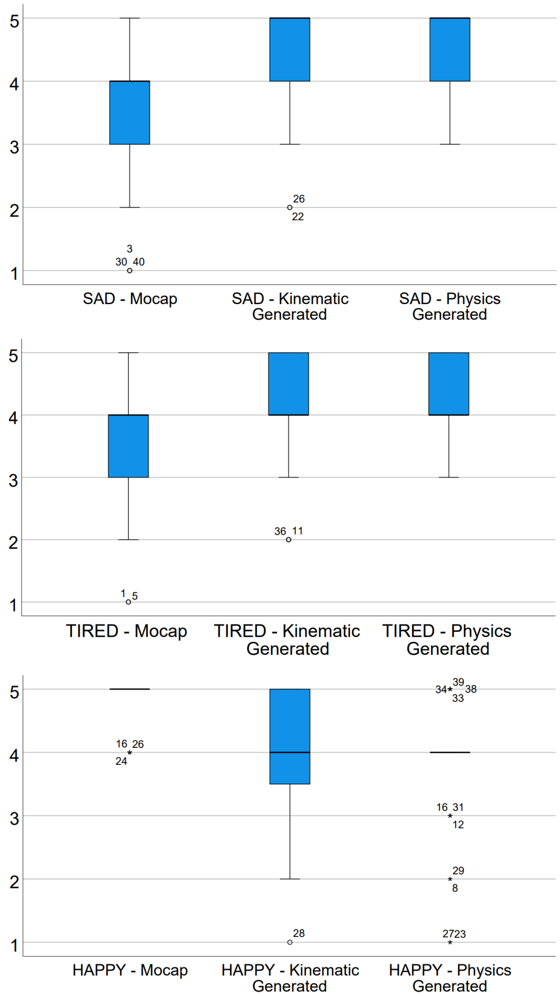

Fig. 12. Boxplot charts showing the value distribution for the emotions "Sad", "Tired" and "Happy" in regards to each type of clip.

GUI that allows users to alter a baseline motion's emotion by specifying new desired PAD coordinates in real time and without the need for any specific domain knowledge. It should be noted, however, that this interface serves only to illustrate the underlying framework's capabilities, which could be used independently in professional applications.

In terms of future improvements, the dataset we used grouped animations into preset styles rather than emotional coordinates. As such, all animations that aimed to express the same emotion were labeled with the exact same Pleasure, Arousal and Dominance values. In reality not all animations with the same emotion express it with the same intensity, and the emotional coordinate values are subject to change even during the course of the same animation. It would be interesting to explore our approaches using an enriched dataset that further split each animation's labels into chunks, adding more granularity to the emotional expression of the data. Increasing the overall animation and emotional variety, alongside further tweaking our motion generation heuristic rules may also improve our system's performance on non-locomotion animations.

References

[1] Bullet real-time physics simulation, https://pybullet.org/wordpress/

[2] Agrawal, S., Shen, S., van de Panne, M.: Diverse motion variations for physics-based character animation. In: Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. pp. 37–44 (2013)

[3] Aristidou, A., Zeng, Q., Stavrakis, E., Yin, K., Cohen-Or, D., Chrysanthou, Y., Chen, B.: Emotion control of unstructured dance movements. In: Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation. pp. 1–10 (2017)

[4] Bandai Namco Research Inc.: Bandai-Namco-Research-Motiondataset (2022), https://github.com/BandaiNamcoResearchInc/Bandai-Namco-Research-Motiondataset

[5] Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)

[6] Coros, S., Karpathy, A., Jones, B., Reveret, L., Van De Panne, M.: Locomotion skills for simulated quadrupeds. ACM Transactions on Graphics (TOG) **30**(4), 1–12 (2011)

[7] Groff, E.: Laban movement analysis: Charting the ineffable domain of human movement. Journal of Physical Education, Recreation & Dance **66**(2), 27–30 (1995)

[8] Hoffmann, H., Scheck, A., Schuster, T., Walter, S., Limbrecht, K., Traue, H.C., Kessler, H.: Mapping discrete emotions into the dimensional space: An empirical approach. 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC) pp. 3316–3320 (2012)

[9] Holden, D., Saito, J., Komura, T.: A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics (TOG) **35**(4), 1–11 (2016)

[10] Izard, C.E.: Basic emotions, relations among emotions, and emotion-cognition relations. (1992)

[11] Lee, Y., Kim, S., Lee, J.: Data-driven biped control. In: ACM SIGGRAPH 2010 papers, pp. 1–8 (2010)

[12] Logan King, B.M.: Bvhtodeepmimic (2019), https://github.com/BartMoyaers/BvhToDeepMimic

[13] Ma, L.K., Yang, Z., Tong, X., Guo, B., Yin, K.: Learning and exploring motor skills with spacetime bounds. In: Computer Graphics Forum. vol. 40, pp. 251–263. Wiley Online Library (2021)

[14] Mehrabian, A.: Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. Current Psychology **14**(4), 261–292 (1996)

[15] Peng, X.B., Abbeel, P., Levine, S., van de Panne, M.: Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. ACM Transactions on Graphics (TOG) **37**(4), 1–14 (2018)

[16] Powell, M.J.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. The computer journal **7**(2), 155–162 (1964)

[17] Randhavane, T., Bhattacharya, U., Kapsaskis, K., Gray, K., Bera, A., Manocha, D.: Identifying emotions from walking using affective and deep features. arXiv preprint arXiv:1906.11884 (2019)

[18] Reynolds, R., Novotny, E., Lee, J., Roth, D., Bente, G.: Ambiguous bodies: The role of displayed arousal in emotion [mis] perception. Journal of Nonverbal Behavior **43**(4), 529–548 (2019)

[19] Russell, J.A.: A circumplex model of affect. Journal of personality and social psychology **39**(6), 1161 (1980)

[20] Safonova, A., Hodgins, J.K.: Construction and optimal search of interpolated motion graphs. In: ACM SIGGRAPH 2007 papers, pp. 106–es (2007)

[21] Tomkins, S.S.: Affect theory. Approaches to emotion **163**(163-195), 31–65 (1984)

[22] Wang, W., Huang, Y., Wang, Y., Wang, L.: Generalized autoencoder: A neural network framework for dimensionality reduction. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 490–497 (2014)

[23] Wang, Y., Yao, H., Zhao, S.: Auto-encoder based dimensionality reduction. Neurocomputing **184**, 232–242 (2016)

[24] Yin, K., Loken, K., Van de Panne, M.: Simbicon: Simple biped locomotion control. ACM Transactions on Graphics (TOG) **26**(3), 105–es (2007)