MEET: The Music Event Emotion Tracking Metaverse

Luigi Atzori[1,2], Gulnaziye Bingol[1,2], Concetta Cantone[3], Nicola Conci[4], Matteo Fasa[3], Alessandro Floris[1,2], Giulia Martinelli[4], Marina Samarotto[3], and Salvatore Serrano[5,6]
[1]DIEE, University of Cagliari, 09123 Cagliari, Italy. [2]CNIT, University of Cagliari, 09123 Cagliari, Italy
[3]Xenia Progetti Srl, 95021 Aci Castello, Catania, Italy. [4]DISI, University of Trento, 38123 Trento, Italy
[5]Department of Engineering, University of Messina, 98166 Messina, Italy
[6]CNIT, University of Messina, 98166 Messina, Italy
{gulnaziye.bingol, l.atzori}@unica.it, ccantone@xeniaprogetti.it, nicola.conci@unitn.it, MFasa@xeniaprogetti.it, alessandro.floris84@unica.it, giulia.martinelli-2@unitn.it, msamarotto@xenianetworksolutions.it, sserrano@unime.it

***Abstract*—This paper presents the Music Event Emotion Tracking (MEET) Metaverse, which is one of the demo results produced within the FUN-Media project. The MEET Metaverse is a virtual disco where multiple users can join together through their avatars to enjoy musical events. The peculiar characteristic of the MEET Metaverse is that the emotions of participants are inferred from their facial expressions and speech, and are used to select the next song to be played in the disco based on the average emotional state of participants. Moreover, avatars' facial poses are updated based on participants' emotions, and realistic avatar animations are reproduced using a combination of motion retargeting and high-fidelity appearance modeling.**

***Index Terms*—Metaverse, Virtual Reality, Facial Emotion Recognition, Speech Emotion Recognition, Avatar Rendering.**

I. Introduction

The Metaverse offers a novel experience to users by opening the doors to social-based, multi-user environments, merging physical reality with digital virtuality [1]. Metaverse applications have been developed in different areas, such as Smart City, gaming, tourism, culture, and education [2], [3].

In this paper, we present the Music Event Emotion Tracking (MEET) Metaverse demo, a virtual disco environment where people can meet through their avatars to join musical events. The MEET Metaverse provides the means to communicate via WebSocket through external systems that implement Facial Emotion Recognition (FER) and Speech Emotion Recognition (SER) algorithms. The demo will show how multiple people can join the virtual disco of the MEET metaverse (shown in Fig. 1) wearing a Meta headset. The estimated participants' emotions will be used to update the avatar's facial poses and to drive song selection within the disco room according to the average emotional state of all participants.

The next sections describe the system architecture, the implemented FER and SER algorithms, and the proposed approach for the rendering of avatar movements.

This work has been supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP C29J24000300004, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

Fig. 1: The MEET metaverse.

II. System Architecture

Fig. 2 shows the system architecture of the MEET Metaverse application, whose two main components are the Business Logic and the Datasets modules. The Business Logic includes the implementation and set of all logic and functionalities that comprise the application, such as user interactions within the virtual environment or general user interface interactions. The Business Logic receives the estimated emotions from the users and uses the data from the Datasets module to implement the necessary features and functionalities concerning the next song selection and next pose avatar selection.

The MEET Metaverse application runs on headsets belonging to the Meta Quest family; therefore, it is usable through a GUI (Graphical User Interface) that serves as an interaction point between users and the metaverse environment. The flow of the virtual musical event is based on the following steps: users start the application and choose an avatar; then, to enjoy the musical event, they automatically join the shared VR disco room, which is the meeting point of all participants and their avatars. At this point, through the tracking features provided by the headset, the avatar reproduces all the user's body movements and facial expressions.

During the song playback, external systems, which can be referred to as "emotion providers", infer the emotions of all participants. The emotion providers are located in a logical entity named "Provider Station", which estimates the participants' emotions using both FER and SER algorithms.

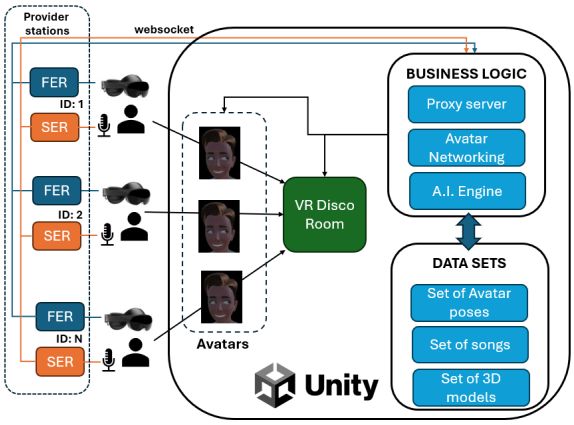

Fig. 2: The system architecture of the MEET metaverse.

When the providers' emotion data is available, it will be sent in JSON format through a WebSocket channel to the MEET proxy server, which forwards the emotion to the correct headset, via WebSocket as well. To apply the emotions to the avatars, body and expression tracking will be turned off for a configured time, and each avatar will take a specific new body and facial pose representing the received emotion. The pose is taken from the Datasets module. Furthermore, the Business logic runs an AI engine that processes the data received from the provider stations and, based on that data, determines the next song to be played in the playlist with the corresponding video clip, and communicates it to all the headsets, which will synchronize their playlist.

III. Facial Emotion Recognition

The FER module implements emotion detection by using Meta's Face Tracking [4] and Eye Tracking APIs [5] to collect facial features and establish their correspondence with the Facial Action Coding System (FACS) taxonomy [6]. The implementation takes advantage of Meta's Face Tracking for Movement SDK capabilities, which provide real-time facial feature tracking data, and maps these features to standardized FACS Action Units (AUs) to ensure psychologically validated emotion classification. The methodological approach builds on the FACS framework, which systematically decomposes human facial expressions into 46 AUs corresponding to anatomically distinct muscle group activations, implementing a structured semantic mapping that establishes bidirectional correspondence between Meta's facial tracking nomenclature and their corresponding standardized FACS identifiers.

The system analyzes seven distinct emotional states: the six basic emotions (happiness, sadness, surprise, fear, anger, and disgust) alongside a neutral state classification [6] when emotion intensities fall below the established threshold. The implementation lies in the development of a bilateral pattern recognition methodology that accounts for the inherent tolerance to asymmetry in natural facial expressions, employing grouped AU pattern structures where each emotion is defined by multiple AU clusters, with recognition contingent on the activation of at least one variant (left or right) within each cluster [7]. The recognition framework operates where each emotion E_i is characterized by a collection of AU clusters $C_1, C_2, ..., C_n$, with each cluster C_j containing bilateral AU variants (e.g., CheekRaiserL and CheekRaiserR for the same facial movement). The activation condition for emotion recognition is formally expressed as:

$$\forall j \in \{1, 2, \ldots, n\}, \exists AU_{jk} \in C_j : w(AU_{jk}) > \theta_{\min} \quad (1)$$

Eq. (1) defines the activation condition where for all clusters j, there must exist at least one AU variant AU_{jk} within cluster C_j such that its weighted intensity $w(AU_{jk})$ exceeds the minimum activation threshold $\theta_{\min}$. This formulation ensures that while all anatomically distinct facial movements required for an emotion must be present, the system accommodates natural facial asymmetry by requiring activation of only one side per movement type. The emotion intensity calculation employs a cluster-weighted averaging scheme:

$$I(E_i) = \frac{1}{|C|} \sum_{j=1}^{|C|} \max_{k \in C_j} w(AU_{jk}), \quad (2)$$

where $I(E_i)$ represents the intensity of emotion E_i, calculated as the arithmetic mean of the maximum activation weights observed within each required AU cluster C_j. The term $|C|$ denotes the total number of clusters required for emotion E_i. The final classification stage employs probabilistic normalization across all candidate emotions: $\sum_{i=1}^{N} I(E_i) = 1$, where N represents the total number of emotions considered. This equation ensures that the sum of all emotion intensities equals unity, facilitating interpretable confidence scores and allowing neutral state detection when the intensity of the dominant emotion remains below user-defined sensitivity parameters. The temporal processing pipeline implements aggregation across one-second intervals to mitigate sensor noise and transient micro-expression artifacts commonly encountered in real-time facial tracking systems. The emotion classification result is then formatted as a JSON payload, including the emotion label, a numeric code, a timestamp, and the station ID, and it is sent to the MEET proxy server.

IV. Speech Emotion Recognition

The Speech Emotion Recognition model (SER) leverages Wav2Vec2, a self-supervised learning architecture originally designed for speech representation learning [8], which combines convolutional neural networks (CNNs) for raw waveform feature extraction and transformer encoders [9] for contextual representation learning. Specifically, the proposed architecture fine-tunes the Italian-optimized variant jonatasgrosman/wav2vec2-large-xlsr-53-italian [10], which extends the XLSR-53 cross-lingual model [11], [12] to Italian. A sequence classification head is appended to map the transformer's output to emotion labels, dynamically adjusted based on the target dataset's emotion classes. The training pipeline processes raw audio using *Wav2Vec2Processor* to normalize waveforms and generate input features compatible

TABLE I: Speech Emotion Recognition performance.

	Precision	Recall	F1-score	Support
Guilt	0.90	0.83	0.87	1005
Happiness	0.81	0.81	0.81	1643
Fear	0.76	0.80	0.78	927
Surprise	0.94	0.85	0.89	1017
Sadness	0.72	0.80	0.76	740
Anger	0.78	0.77	0.78	848
Disgust	0.64	0.69	0.67	752
Neutral	0.71	0.70	0.70	846
Accuracy		0.79		7778
Macro avg	0.78	0.78	0.78	7778
Weighted avg	0.79	0.79	0.79	7778

with the pre-trained encoder. Emotion labels are converted to numeric IDs, decoupling the model from dataset-specific label semantics. Training is conducted using Hugging Face's Trainer framework with 5 epochs and a batch size of 4.

The model training was performed on the DEMoS speech corpus audio signals [13], which were sampled using a sampling rate $F_s = 44100$ Hz, 1 channel, and a precision of 16-bit per sample. All signals were downsampled to $F_s' = 16000$ Hz by means of *torchaudio.transforms.Resample*. The resulting speech signals were split into overlapped windows lasting $W_l = 500$ ms. Each window overlaps the next one for an interval $W_o = 400$ ms (i.e., we move the sliding window with a step equal to $W_s = 100$ ms). The obtained audio raw waveforms are the input of the *Wav2Vec2Processor*, which normalizes waveforms and generates input features compatible with the pre-trained encoder. Table I reports the classification performance obtained on the test subset of the DEMoS speech corpus, using a 60%/20%/20% split of the selected prototypical samples for training, validation, and test, respectively.

The implemented SER system uses the *websockets* library to establish and manage asynchronous WebSocket connections with multiple clients. These clients send JSON-encoded messages containing chunks of audio data encoded as NumPy float32 arrays along with metadata such as the sampling rate and station ID. Upon receiving audio, the server writes the data into a circular buffer and performs a root mean square (RMS) check to filter out segments with low signal energy, which are classified as "No Signal" to avoid unnecessary computation. The circular buffer is initialized using a NumPy array of zeros with a fixed size. Specifically, the buffer duration is 0.5 seconds. The buffer is designed to retain the most recent half-second of audio, continuously updated as new audio chunks arrive. This circular mechanism supports real-time audio processing without data loss or excessive memory use. For segments that pass the RMS threshold, the script processes the audio using the trained model.

For emotion classification, inference is performed over the full 0.5-second buffer to provide sufficient temporal context for the Wav2Vec2 model, consistent with the chunk duration used during training. This combination of a sliding window approach and circular buffering ensures that audio is processed efficiently and with minimal latency, allowing the system to react quickly to changing emotional states in the incoming speech signal. The corresponding *Wav2Vec2Processor* prepares the audio data by converting it into a suitable format for the model, returning PyTorch tensors. These are then passed to the model, which produces logits that are converted into predicted emotion classes using *torch.argmax*. The predicted class index is finally decoded into a human-readable label using a *LabelEncoder*. The classification result is then sent to the MEET proxy server. The message, formatted as a JSON payload, includes the emotion label, a numeric code, a timestamp, and the station ID. The server handles all of this asynchronously using Python's asyncio framework, enabling it to scale efficiently with multiple concurrent audio streams.

V. Rendering of Avatar Movements

To enable realistic and physically coherent avatar animation within the virtual environment, we adopt a modular approach that combines motion retargeting and high-fidelity appearance modeling. The core animation engine employs a transformer-based architecture [14] trained with masked pose modeling to learn a unified motion representation across heterogeneous skeletal structures. Human motion is typically captured using joint-based, rigid kinematic skeletons [15], [16], yet these vary widely across datasets and acquisition systems. This variability, alongside the scarcity of reliable 3D ground-truth annotations in uncontrolled environments, makes generalized motion learning a challenge.

Our approach addresses this by first encoding motion sequences via a skeleton-aware autoencoder that masks random joints in both space and time. A transformer model then reconstructs the full pose using a shared reference skeleton, enabling unpaired motion transfer across non-homeomorphic skeleton topologies. In the second stage, a shape-aware optimization refines the motion at the mesh level. Specifically, a face-based collision minimization strategy is introduced to prevent mesh interpenetration during animation, improving on traditional vertex-based techniques by resolving intersections more precisely on the mesh surface. The final retargeted skeleton is bound to the target mesh via Linear Blend Skinning (LBS), while joint positions are refined to ensure anatomically plausible deformations.

To complement motion realism with visual fidelity, we integrate detailed appearance modeling by collecting a volumetric dataset of human performances, which enables high-resolution reconstruction of real human bodies, clothing, and facial details. The resulting 3D meshes are textured, rigged, and synchronized with the motion-retargeted skeletons. This allows the generation of photorealistic animated avatars that move naturally while preserving identity and clothing geometry.

VI. Conclusion

This paper presented the MEET Metaverse demo, a virtual disco environment where people can meet through their avatars to join musical events. FER and SER algorithms have been developed to infer emotions from participants, which are used to update the avatar's facial poses according to participants' emotions, and to drive song selection within the disco room.

[1] S. Mystakidis, "Metaverse," *Encyclopedia*, vol. 2, no. 1, pp. 486–497, 2022.

[2] S.-M. Park and Y.-G. Kim, "A Metaverse: Taxonomy, Components, Applications, and Open Challenges," *IEEE Access*, vol. 10, pp. 4209–4251, 2022.

[3] H. Wang, H. Ning, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand, "A Survey on the Metaverse: The State-of-the-Art, Technologies, Applications, and Challenges," *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14 671–14 688, 2023.

[4] Meta, https://developers.meta.com/horizon/documentation/unity/move-face-tracking/, accessed: 2025-04-03.

[5] ——, https://developers.meta.com/horizon/documentation/unity/move-eye-tracking/, accessed: 2025-04-03.

[6] P. Ekman and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. San Francisco: Consulting Psychologists Press, 1978.

[7] "emo_HuTwin: Emotion recognition system," https://github.com/gulnazbingoll/emo_HuTwin, 2025, accessed: 2025-06-03.

[8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] J. Grosman, "Fine-tuned XLSR-53 large model for speech recognition in Italian," https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-italian, 2021.

[11] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised Cross-lingual Representation Learning for Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2426–2430.

[12] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "Xls-r: Self-supervised cross-lingual speech representation learning at scale," 2021. [Online]. Available: https://arxiv.org/abs/2111.09296

[13] E. Parada-Cabaleiro, G. Costantini, A. Batliner, M. Schmitt, and B. W. Schuller, "Demos: an italian emotional speech corpus: Elicitation methods, machine learning, and perception," *Language Resources and Evaluation*, vol. 54, no. 2, p. 341–383, Feb. 2019.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[15] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021.

[16] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," 2019. [Online]. Available: https://arxiv.org/abs/1904.04514