GANs and Poses: An Interactive Generative Music Installation Controlled by Dance Moves

Richard Vogl*†
Faculty of Informatics
TU Wien
Vienna, Austria
richard.vogl@tuwien.ac.at

Hamid Eghbal-zadeh*
Institute of Computational Perception
JKU Linz, Austria
hamid.eghbal-zadeh@jku.at

Gerhard Widmer
Institute of Computational Perception
JKU Linz, Austria
gerhard.widmer@jku.at

Peter Knees
Faculty of Informatics
TU Wien
Vienna, Austria
peter.knees@tuwien.ac.at

* Equal contributions.
† Richard Vogl is also affiliated with the Institute of Computational Perception at JKU Linz, Austria.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

Abstract
GANs and Poses is an interactive multi-media installation using human poses to control artificial neural networks generating parts for a music arrangement. To provide incentive for interaction, a game-style UI rewards the audience with audio-visual feedback, which in turn drives the progress through the musical arrangement. This work uses a variety of artificial neural networks to (i) detect persons and their body posture, (ii) classify dance moves, and (iii) generate drum as well as melody patterns.

Introduction
In the recent past, generative artificial neural networks (NN) have gained popularity and received much attention in science as well as popular media. Particularly, Generative Adversarial Neural Networks (GANs) [2] and Recurrent Neural Networks (RNNs) are successfully applied for generating high resolution images [7, 8], audio [1], and music [9] with surprising quality. While first attempts to generate music with neural networks resulted in compositions lacking structural consistency, recent demonstrations are more convincing and show the potential of these methods [5].

In this work, we combine state-of-the-art NN-based human pose estimation with an experimental generative music creation method utilizing GANs with a recurrent-convolutional architecture. Using these technologies, a semi-human/semi-
machine music-ensemble is created when the audience interactively controls song structure and generated musical elements using body gestures. To encourage user engagement, a simple game-style reward system is incorporated in the system. By using recognized dance moves as achievements and the individual parts of the music arrangement’s structure as game levels, the UI presents incentives for users to interact with the system. Users can score points by performing various dance moves which results in visual and musical effects and progresses the music through the different levels/parts of the arrangement.

On an artistic level, a musical composition is driven by the engagement and expression of individual actors, each contributing their skills to a joint progression of a musical experience. Capturing multiple performers and their moves adds a social dimension to the installation, stimulating passive spectators and turning them into active performers, motivated to add to the generation of sound.

System
An overview of the system can be found in Figure 2. As mentioned, the input for the interactive installation is provided by a real-time multi-person pose recognition system. This system is implemented using a 12-layer convolutional neural network (CNN) designed to be efficient during inference (MobileNets [4]). The implementation follows the open-pose approach to real-time pose detection [10] and was trained on the MSCOCO Keypoints dataset [6]. Figure 1 shows the input and detected body parts of the pose estimator.

Using the detected stick-figure-like coordinates extracted from the input video, a dance move recognition system is used to detect if any of the present persons are dancing. To this end, videos of character dances from the video game *Fortnite* were used as training data. The videos of the game character dances represent clean samples of 26 different dances, and were chosen for their easy accessibility and volume of good samples of dance moves. Figure 3 shows samples of the screenshots for different dances of the collection. The input data for training was extracted using the pose estimation system, using additional data augmentation to increase robustness. A two-layer Long Short-Term Memory (LSTM) [3] neural network was trained using the 1.5 second sequences of the predicted pose coordinates as inputs and the 26 dance moves as output classes.

For both melody and rhythm generation, a convolutional-recurrent GAN architecture was used. The GANs were

---

1 [http://cocodataset.org](http://cocodataset.org)

2 [https://www.youtube.com/watch?v=2XtLEp0XAll](https://www.youtube.com/watch?v=2XtLEp0XAll)
trained in a semi-supervised setup to allow parametrization
of generated patterns, which is used to control complexity
of the generated music to match the different parts of
the music arrangement. As training data, a MIDI dataset ex-
tracted from a freely available online collection\textsuperscript{3} was used.

A simple game-engine and user interface (UI) implemented
in openFrameworks\textsuperscript{4} represents the centerpiece of the sys-
tem. As can be seen in Figure 4, the UI consists of a vi-
sualization of the detected poses, the current part/level of
the musical arrangement represented by stars, as well as
bars which indicate remaining time until a certain amount
of dances has to be performed to achieve further progress.
In Figure 4 we provide different screenshots of the UI while
two persons are interacting with the installation. Commu-
nication with the pose detection software as well as with
the pattern generation server is implemented using Open
Sound Control\textsuperscript{5}.

While the drum part and lead melody is generated using
GANs, the installation uses a predefined arrangement pro-
viding basic chord progressions for each of the levels-parts
of the musical piece. This arrangement uses a live-set com-
piled in Ableton Live\textsuperscript{6}, which is controlled via MIDI by the
openFrameworks application.

Interaction
The installation shows a screen with stars representing the
different stages of the music arrangement (e.g.: intro, verse,
bridge, chorus – see Figure 4). When a person steps in
front of the camera, a stick-figure model of the recognized
body will be shown on screen. The played music and a par-
ticle animation which reacts to the pose and music is sup-
posed to animate the person to move. As soon as dance-
like moves are detected, visual feedback provides more in-
centive to dance. When enough dance moves are detected,
the arrangement moves to the next part, which results in
a change in chord progression, and adds additional ele-
ments such as generated drums as well as melody. When
the arrangement is in a state other than the base-state,
the audience has only limited amount of time to perform
dance moves, until the arrangement falls back into the pre-
vious stage (the timer is indicated with a red bar). Since the
change of segments is synchronized to the chord progres-
sion, a green timer is shown after sufficient dance moves
are detected and indicates when the music will progress
to the next part. A video that demonstrates this installation
can be found at https://youtu.be/mg7_\textsubscript{K}zwIhC8.

Requirements and Setup
The minimum requirements for setting up the installation are:

- An electricity outlet to power the computer the system
  runs on.
- A reasonably well lit space which allows one or more
  persons to stand in front of the webcam in a distance
  of 2-3m.
- A platform to put the computer on, ideally chest height,
to allow good visibility of the screen while providing a
natural perspective of the webcam.

The setup is straightforward since all the components of
the installation can be run on a MacBook Pro with required
GPU, webcam, and screen built in. We can bring and pro-
vide the laptop with pre-installed software for the duration
of the conference. Additionally required headphones and a

\textsuperscript{3}http://www.midiworld.com
\textsuperscript{4}https://openframeworks.cc
\textsuperscript{5}http://opensoundcontrol.org/
\textsuperscript{6}https://www.ableton.com/en/live
Figure 4: Screenshots of the installation in action. Image a shows the UI with additional development debug information such as the song state (upper right corner) and generated drum and melody patterns (lower half of screen). Image b shows a screen of the clean, camera ready UI without debug information.

headphone amplifier (to allow multiple people hearing the audio) can also be provided by us.

For best presentation and UX, the following additional components are desirable, however, not essential for a successful setup: (i) a big screen and external webcam (HDMI and USB respectively) (ii) external speakers, and (iii) space for more than 2 people.

Acknowledgements
This work has been partly funded by the Austrian FFG under the BRIDGE 1 project SmarterJam (858514), as well as by the Austrian Ministries BMVIT and BMWFW, and the Province of Upper Austria, via the COMET Center SCCH.
REFERENCES


