TAKE THE LEAD:TOWARD A VIRTUAL VIDEO DANCE PARTNER

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Ty Farris

August 2021

COMMITTEE MEMBERSHIP

TITLE: TAKE THE LEAD:TOWARD A VIRTUAL VIDEO DANCE PARTNER

AUTHOR: Ty Farris

DATE SUBMITTED: August 2021

COMMITTEE CHAIR: Jonathan Ventura, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Franz Kurfess, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Paul Anderson, Ph.D.
Professor of Computer Science

ABSTRACT

TAKE THE LEAD:TOWARD A VIRTUAL VIDEO DANCE PARTNER

Ty Farris

My work focuses on taking a single person as input and predicting the intentional movement of one dance partner based on the other dance partner's movement. Human pose estimation has been applied to dance and computer vision, but many existing applications focus on a single individual or multiple individuals performing. Currently there are very few works that focus specifically on dance couples combined with pose prediction. This thesis is applicable to the entertainment and gaming industry by training people to dance with a virtual dance partner.

Many existing interactive or virtual dance partners require a motion capture system, multiple cameras or a robot which creates an expensive cost. This thesis does not use a motion capture system and combines OpenPose with swing dance YouTube videos to create a virtual dance partner. By taking in the current dancer's moves as input, the system predicts the dance partner's corresponding moves in the video frames.

In order to create a virtual dance partner, datasets that contain information about the skeleton keypoints are necessary to predict a dance partner's pose. There are existing dance datasets for a specific type of dance, but these datasets do not cover swing dance. Furthermore, the dance datasets that do include swing have a limited number of videos. The contribution of this thesis is a large swing dataset that contains three different types of swing dance: East Coast, Lindy Hop and West Coast. I also provide a basic framework to extend the work to create a real-time and interactive dance partner.

ACKNOWLEDGMENTS

Thanks to:

- Professor Ventura for being my advisor and supporting me along the way.
- Professor Kurfess and Professor Anderson for being on my committee.
- My mom and dad for loving and encouraging me throughout my thesis.
- My grandfather who impacted my education and will always live on in my memories.
- Professor Smith for being my academic dad and always giving me advice along the way.
- Laura for supporting me and being the best roommate I could ever ask for.
- Kyle for always making me smile and laugh through the high and low moments.
- Olivia for believing in me and supporting me from a million miles away.
- Sean for always helping me out during the challenging times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Recently, the entertainment and gaming industries are trying to combine dance and computer vision. Some of the existing dance applications are creating a virtual dance partner, virtual dance instructor, and special effects based on dance movements. My thesis contributes a survey of computer vision and dance applications. The following topics are covered: existing pose movement models, dance datasets and dance applications.

My thesis further contributes a swing dance dataset that contains the following dance types: East Coast Swing, West Coast Swing and Lindy Hop. Pulling the videos from YouTube, my dataset contains 712 video clips that contain one dance couple and clear backgrounds. Using OpenPose to identify the body parts, this dataset focuses on beginner level instructional swing dance videos for clear and precise dance movements.

Currently there are no datasets that only focus on swing dance, and the few existing datasets that do include swing dance only contain a small amount of videos. Furthermore, there are very few existing dance applications that focus on a dance couple. The related works mostly use motion capture and virtual reality environments to recreate the virtual dance partner, but my work is cost effective since my dataset is only comprised of YouTube videos.

Using this swing dataset, my thesis attempts predict the opposing partner's dance movements based on the input of the current partner's position. In a dance couple, there are two roles: a lead and a follow. If the current dancer is the lead then

my approach will predict the follow and vice versa. I apply 3 different models to predict the opposing dance partner's movements: linear regression model, K-Nearest Neighbors regression model, and Extra Trees regression model.

This problem is challenging because human pose estimation can struggle with identifying people in close proximity. Furthermore, there is no existing work that attempts this challenge specifically towards swing dance. My contributions provide a basic framework to create a virtual dance partner that could be displayed and interacted with in real-time.

Regarding the rest of my thesis, chapter 2 includes background information about human pose estimation approaches and the models used during this experiment. Chapter 3 covers existing pose movement models, dance datasets and dance applications. Chapter 4 describes the features of the swing dataset. Chapter 5 explains the approach taken to predict the dance partner's corresponding movement based on the current dancer's position in a video frame. Chapter 6 shows and analyzes the quantitative and qualitative results. Chapter 7 provides a conclusion and chapter 8 includes future steps.

CHAPTER 2

BACKGROUND

Pose Estimation Approaches

There are two main approaches to pose estimation model architectures: top-down or bottom-up. Top-down method uses a bounding box object detector to identify the individuals in an image and then attempts to infer the body poses. The main problem with top-down methods is early commitment because when the detection of individuals fail, there is no possibility of recovering [32]. In terms of computational cost and run-time, top-down approaches are proportional to the number of people in the image because a single-person pose estimator is run for every detection.

Instead the bottom-up method identifies the local body parts and then groups the entities together into a person. This approach is robust to early commitment and presents a detached run-time complexity from the number of people in an image [6]. This method does not directly use global contextual cues from other body parts, but can be incorporated. The main challenge with bottom-up approaches is grouping body parts when there is a large overlap between people [32].

Models Overview

The linear regression model is a simple implementation that is mainly used to find the relationship between the variables or for forecasting [17]. This model is made to predict target values based on independent variables. Some disadvantages are that linear regression is greatly affected by outliers and cannot capture the real world complexities because of the assumption of a linear relationship between independent and dependent variables [39].

K-Nearest Neighbors regressor is another simplistic model that uses the K-Nearest Neighbors algorithm to predict a target based on the nearest neighbors in the training set[16] [12]. Compared to the linear regression, this algorithm is non-parametric which means that there are no assumptions for the data. The algorithm is also sensitive to outliers and k-nearest neighbors loses efficiency as the dataset size increases [18].

Extra trees regressor implements a tree-based ensemble method which strongly randomizes both attribute and split partition at a tree node[19] [4]. This regression model fits a set of randomized decision trees on subsets of the data and uses averaging to improve the prediction. This algorithm adds randomization but still provides optimization [3].

CHAPTER 3

RELATED WORK

3.1 Pose Movement Models

ConvNet

ConvNet uses a top-down approach and combines a deep convolutional network (ConvNets) and graphical model. The network uses multiple resolution CNN architectures in parallel to create a pose estimation for a single person. This system creates discrete heat-maps to predict the probability of the location of the individual joints in images [32]. ConvNet contains 3 modules: coarse heat-map localization, sampling and cropping features for the joints, and fine-tuning.

CPM

CPM (Convolutional Pose Machines) combines convolutional neural networks with pose machines to create a top-down method for single person pose estimation. This model architecture produces a belief map to refine body part location estimates [50]. By using a sequential prediction framework, CPM learns the image features and image-dependent spatial models. Using a VGG structure at each stage, the first stage predicts the belief maps for the keypoints in the original image. The subsequent stages predicts the keypoints from the belief map in the previous stage and the result of the original image passed through a VGG neural network [6].

DeepCut

DeepCut uses a bottom-up approach that jointly infers the number of people, their poses, spatial proximity, and part level occlusions [34]. Detection and pose estimation are commonly separated steps, but the proposed model combines subset partitioning and labeling. The system first detects all the body parts in an image and then jointly clusters and labels the body parts to each corresponding person in the image. This system can successfully identify the number of people per image and the number of visible body parts per individual.

DeepPose

DeepPose is a single person pose estimation model that uses a top-down approach. This model was the first significant research article that uses AlexNet as a backbone architecture and applied deep neural networks (DNN) to human pose estimation [32]. By using a 3-stages cascade of DNN regressors, the system is able refine coarse poses to create high precision and pose estimation [47].

OpenPose

OpenPose is the first open-source real-time system for multi-person 2D pose detection that achieves high accuracy [6]. This system uses a bottom-up approach and Part Affinity Fields (PAFs) to identify body, foot, hand and facial keypoints. PAFs are 2D vector fields that represent the position and orientation of the body parts. OpenPose is widely used today and has been included in the OpenCV library.

Stacked Hourglass Network

The stacked hourglass network stacks down-sampling and up-sampling layers on different scales to capture different spatial relationships features, such as body orientation or body part arrangements. The architecture combines bottom-up and top-

down approach with intermediate supervision to improve the model performance [33]. By using max pooling for down-sampling and nearest-neighbor interpolation for up-sampling, this system captures global and local information. In the Stacked hourglass network, both high-resolution to low-resolution processing and low-resolution to high resolution processing are symmetrical [6].

Overview

There are many different pose estimation model architectures that use convolutional neural networks, VGG, deep neural networks, and stacked hourglass network. The models mainly use a top-down or bottom up approach for pose estimation. When identifying a person's body joints, some models include heat map localization while other models focus on the spatial proximity and spatial relationship features. Since OpenPose is commonly used and provides an OpenCV library, I used this system to identify the skeleton keypoints in the swing dance videos.

3.2 Dance Datasets

3.2.1 Salsa

UCF101 - Action Recognition Data Set

This dataset contains action videos that classified into 101 categories collected from YouTube [41]. UCF101 is known to have the largest diversity and most realistic action videos. There is a total of 13,320 videos, but the videos are combined into 25 groups. Each group has 4-7 videos that share common features, such as similar viewpoint. Salsa spins are one of the categories within UCF101 and are typically 5-10 seconds. These clips are saved in .avi files and have a resolution of 320x240.

Dance DB Motion Capture Database

By incorporating motion capture technology, this database contains local and traditional dance videos, such as Greek and Cypriot dances [44]. The motion capture suit has 38 markers and the motion capture system contains 24 cameras. Depending on the dance, the clips are saved in a couple of different formats: AutoDesk FBX, C3D motion capture data and MP4 videos. Regarding salsa, the database contains two datasets, and each dataset contains 3D motion capture data for one performer.

3.2.2 Tango

HDM12 Dance Dataset

This database contains Argentine Tango dance sequences for 11 different dance couples [49]. There is 149 motion clips that range between 21 and 78 seconds, and in total there is 97.48 minutes of content. The data is saved as point-cloud trajectories in the formats: C3D and ASF/AMC format. There are 12 cameras and each performer wears 46 retro-reflective markers. The dance couple consists of a man as the lead and a female as the follow.

Exploring Dance Movement Data Using Sequence Alignment Methods

This paper includes a motion capture dataset and video dataset for samba and tango dancers [10]. For samba dances, there are 92 time intervals recorded that record 6 features per person. This performance uses 3 samba dancers, one teacher and two students. For tango dances, there are a couple professional and beginner dancers that had 25 body parts recorded for each person.

Ballroom Dance Dataset

The dataset contains wearable sensor data, video and body keypoints for ballroom dances [30]. There are 7 male dancers that each wear 6 sensors and performs and 13 ballroom dance steps. There is over 100 minutes worth of data that is saved in Full HD (1920×1080). Each dancer is recorded 10 times from the front and 10 times from the back. The wearable sensor data is saved in csv files and the videos are saved in mp4 which are about 50 seconds each. The keypoints are saved in json files from using OpenPose.

Music to Dance Motion-Synthesis

This dataset contains audio files and skeleton points for four types of dance: Cha-cha, Tango, Rumba and Waltz [46]. Overall, there is 94 minutes of motion captured data for the four types of dance. Related to Tango, there are 9 dance recordings that are performed by professional dancers. For each performer, 21 joints are recorded. The skeleton points are saved in json files where the frame rate is 25 FPS (Frames Per Second).

3.2.3 Ballet

Action Recognition by Learning Mid-level Motion Features

The ballet dataset is created from an instructional Ballet DVD [15]. This dataset is publicly unavailable and has 44 sequences that combines dance moves from 8 different actions. In total there are 2 male performers and 1 female performer. Only one dancer is performing in a frame and the frame size is 50x50.

An Efficient Volumetric Framework for Shape Tracking

This paper creates a hybrid multi-camera and marker-based capture dataset [1]. The paper tests the proposed method on multiple datasets, including ballet. The ballet dataset is recorded in full HD (1920x1080) and contains various moves, such as different levels of difficulty or fast moves. There are 9 viewpoints and 500 frames captured in a dance sequence.

Recognizing Action at a Distance

There is a ballet dataset that includes 16 different ballet actions that are choreographed [14]. These actions are taken from an instructional video that includes professional dancers. The ballet performers are composed of two men and two women. In total, there are 24800 frames and each action had 51 frames.

An Approach to Ballet Dance Training through MS Kinect and Visualization in a CAVE Virtual Reality Environment

There are two ballet datasets that represent two different ballet dance sequences [26]. Each dataset has a teacher and a student performing 6 different gestures. The first dataset includes 554 frames and the second dataset includes 1094 frames. The datasets are created using the Microsoft Kinect camera system in order to track skeletal joints.

Automatic Labanotation Generation, Semi-automatic Semantic Annotation and Retrieval of Recorded Videos

The ballet dataset is manually labeled with semantic annotations and focuses on a single performer per frame [13]. There are 83 videos and 22 semantic annotations. The semantic annotations are split into 14 dynamic dance sequences and 8 static dance poses.

Development of a Human Activity Recognition System for Ballet Tasks

The ballet dataset is composed of video and sensor data [21]. There are 23 female pre-professional dancers from a university dance institution. Each dancer wears 6 sensors and takes approximately 45 minutes for data collection. In groups of 2 to 5 people, the dancers perform discrete movements within classical ballet. In total, there are 11 dance actions that can be classified into leg lifting tasks and jumping tasks.

3.2.4 Ballroom

MPII Human Pose Dataset

Containing around 25,000 images, this dataset includes over 40,000 people and annotated body joints [2]. There are 410 human activities and each image was taken from a YouTube video. There are up to 10 videos for each activity that exclude low quality or videos that do not contain people. Related to ballroom, the dataset contains 856 ballroom dancing images. The frames range from containing one person to multiple people. These people could either be the main dancers in the scene or background individuals. The dataset attempts to provide different people in the video or the same person with different poses. The dataset contains a wide variety of human poses, clothing types, and environments.

Ballroom Dance Step Type Recognition by Random Forest Using Video and Wearable Sensor

The ballroom dataset includes video and wearable sensor data [29]. There are 7 male dancers that range from 1 year to 17 years of experience. The performers wear 6 sensors and dance a sequence of 13 steps. The video is recorded on a SONY FDR-

AX60 with a frame rate of 120 FPS(Frames Per Second). For each performer, the first half of the video is shot from one viewpoint and the second half of the video is shot from a second viewpoint. When collecting the data, each dancer will perform 20 times. After running OpenPose, the video data is saved in a JSON file and the sensor data is saved in a CSV file.

Ballroom Dataset

The dataset is a collection of ballroom dance styles where the audio format is labeled with tempo values [20]. The data is collected from BallroomDancers.com which provides ballroom dancing lessons. There are 698 music excerpts that are each around 30 seconds. The tempo ranges between 60 and 224 beats per minute. There are 8 different styles that are recorded within ballroom, such as Rumba and Slow Waltz.

Extended Ballroom Dataset

This dataset is an extension of the Ballroom Dataset and contains 4,180 tracks which have better audio quality [28]. There are about 444 tracks per dance style and an additional 5 classes, such as Foxtrot. Each track has the following annotations: tempo, rhythm class, artist, song title and album name.

3.2.5 K-Pop

Real-time Dance Evaluation by Markerless Human Pose Estimation

This database contains 100 popular K-Pop dances which can be split up into teacher and learner dance sequences [24]. Overall there are 100 teacher sequences and 400 learner sequences. The teacher dataset was recorded using a motion capture system, while the learner dataset used a Microsoft Kinect 2 camera. For the learner database,

there are 4 dancers of various skill levels. These learner dance sequences were labeled one of the following ratings: best, good, bad, and worst.

3.2.6 Street Dancing

AIST Dance Video Database

This database contains 13,939 dance videos that span 10 dance genres and 60 pieces of dance music [48]. Consulted by dance experts, the genres were split between old school styles and new school styles. Each genre has 1,380 videos and 4 categories: basic dance, advanced dance, group dance, and moving camera. Basic and advanced dance sequences cover solo dancing and group dancing. There is an additional 49 situation videos which are categorized as showcase, cypher, or battle. The 40 professional dancers are comprised of 25 male and 15 female and had more than 5 years of dance experience. There are at most 9 cameras to record the dance from different angles which create 118.2 hours of content. The videos were recorded in color and all camera positions were fixed except for the category moving camera.

3.2.7 Combination of Dance Datasets

Let's Dance: Learning From Online Dance Videos

This dataset has 1000 dynamic dance videos and 10 categories [7]. Some of the categories include Swing, Ballet, Tango, and Break Dancing. Each class has 100 videos that are 10 seconds at 30 frames per second. The videos are pulled from YouTube with a quality of 720p. The video clips include dancing performances and plain-clothes practicing.

The Kinetics Human Action Video Dataset

This dataset has 400 human action classes with at least 400 video clips per action [23]. The classes are split into train, test and validation datasets. For each class, there are 250 to 1000 train videos, 100 test videos, and 50 validation videos. Each clip is around 10 seconds and pulled from YouTube. Only one clip is taking from each video and the clips are verified with the correct action using Amazon Mechanical Turkers (AMT). Related to dance, there are 18 different classes that the dataset provides. For example, there are 1114 video clips for tango, 1148 video clips for salsa, and 1144 video clips for ballet. These videos are neither professionally filmed nor edited and production value may vary, such as camera motion, shadows, or a cluttered background. Each video is unique, so there is a variety of dancers, performance, clothing, body pose, shape, age, video resolution, frame rate, camera framing and viewpoint.

3.2.8 Datasets Overview

Overview

One challenge with salsa is that the dancers will move fast so many frames in the video could be blurry. There is a lot of small and subtle hip movements which could be difficult to identify body joints. Tango focuses on sharp and dramatic movements which are easier to identify, but the females often were long dresses that reach the calf or ankle. These long outfits could make the body joints more obscure and difficult to identify. Ballet basics are often easy to identify positions and the outfits do not obscure the body joints. The main problem is ballet videos often focus on one dancer or multiple dancers in synchronous movement. Currently, there are not many ballet dance videos that contain a dance partner.

Ballroom dance is often a collection of dance styles: cha cha, jive, quickstep, rumba, samba, tango, viennese waltz and slow waltz [28]. The mixture of dance types makes identifying dance positions more difficult to classify and identify because there is no consistent set of dance moves that belongs to one style of dance. K-Pop dance typically has no basic steps because K-pop performances will often vary based on the performers and choreographer. The dancers usually wear plain clothes which makes it easier to identify the dance movements. The dance steps are often stiff and rigid positions with a faster tempo to K-Pop songs. This dance style is often performed in group dances that are mostly synchronous.

Street dance is an informal dance style that covers many genres: break, pop, lock, waack, middle hiphop, LA-style hip-hop, house, krump, street jazz, and ballet jazz [48]. This dance style does not have a set of basic moves because this street dancing is free-flowing and creative. The number of performers can vary from one to multiple dancers, and the dances can be synchronous or asynchronous. The performers usually wear plain clothes, but there is typically a large audience or other dancers in the background. This setting could could make it challenging to identify the main performers.

Overall, some datasets focus on specific dance styles, such as the AIST dataset while other datasets have a large collection of dance styles, but very few videos per genre [48]. The existing datasets are typically pulled from YouTube or created with motion capture technology. Sometimes the datasets are not available to the public. Large existing datasets, such as UCF101 and the Let's Dance dataset, often have in the wild data which means that the video content and quality is random [41] [7]. This range of video quality, crowded backgrounds, and dance performers creates a challenge and inconsistency for human pose estimation.

Currently, there is no existing dataset that only focuses on swing dance, and there are very few dance datasets that provide swing dance videos. The Let's Dance dataset is the main existing dataset that includes swing dance, but only has 100 videos. My dataset specifically focuses on swing dance and provides 712 video clips. Most existing datasets focus on single or group performers that typically dance synchronously. My dataset provides a large database of swing dance videos that focus on a single dance couple. In order to minimize the error when identifying the main dance couple in a video frame, my dataset contains no people nor large objects in the background. All the videos are pulled from YouTube channels, and these videos focus on instructional videos for beginners.

3.3 Dance Applications

3.3.1 Interactive Dance Partner

Performance-Driven Dance Motion Control of a Virtual Partner Character

Using a dance motion dataset and a hidden Markov model (HMM), a virtual salsa dance partner is created. The system uses an Oculus Development Kit v2 to capture the user's motion. The dance motion datasets contains motion for both the lead and partner, and the hidden Markov model is trained on the chosen dancer. The chosen dancer becomes the virtual dance partner, and during runtime, the model optimizes and predicts the progress of the chosen dance motion by a forward algorithm [31]. The user can also improvise dance moves because the jump transition allows the model to skip through the dance motion sequence.

Interactive Partner Control in Close Interactions for Real-Time Applications

The virtual character can handle close interactions with a user-controlled character in real time. The framework relies on an interactive mesh which is a spatial relationship-based representation of the body parts [22]. By using a Motion Analysis Eagle Digital optical motion capture system, the interactive partner is created for dancing and fighting scenarios. The virtual partner is created by first matching the user's body keypoints to a pair of poses from the motion library. Then the interaction mesh motion adaption framework edits the selected the dance couple pose and the system renders the virtual dance couple.

Partner Ballroom Dance Robot

A ballroom dance partner robot was developed to research the effectiveness of human-robot coordination [25]. The robot predicts the person's next dance step and generates a dance motion in response. In this scenario, the robot acts as the female dancer and tries to figure out the intention of the lead dancer. The system uses a control architecture based on step transition (CAST) and includes hidden Markov models (HMM)s to create the step estimation.

Daily HRI evaluation at a classroom environment: Reports from Dance Interaction Experiments

This dance robot learned different choreographed dance sequences and can mimic a dance partner in real-time [45]. QRIO is an autonomous robot developed by Sony and is used to create a dance robot in order to study child-robot interaction. The robot can identify the partner's general shape and motion dynamics. This social robot contains the following features: audio recognition, text-to-speech synthesis, visual recognition, short term and long term memory, behavior control architecture and motion control.

QRIO has a playback mode which pulls dance motions that were programmed using a 3-D motion editing system. The robot also has an interactive mode that creates motion imitation using stereo vision cameras.

3.3.2 Dance Training System

A Virtual Reality Dance Training System Using Motion Capture Technology

The dance training system uses motion capture and virtual reality (VR) technologies to record a person's movements, analyze the dance movements and provide feedback [9]. The user's dance movement is recorded by the motion capture system and analyzed by matching the dance position to a motion database. Using 3D animation, the system provides immediate feedback on a dance move by highlighting the body parts that are incorrectly positioned on the student skeleton. A virtual teacher is displayed at the same time to show the correct movement. The system also provides a score report that indicates which body parts are misplaced and may need improvement.

Salsa Dance Learning Evaluation and Motion Analysis in Gamified Virtual Reality Environment

The virtual reality game creates a virtual dance partner to help improve salsa dancing skills. In order to interact with the virtual dance partner, the user wears a VR headset with hand controllers. The virtual dance partner assumes the lead role of the dance couple, so the user acts as the follow. Using inverse kinematics, the optical motion capture system records and links the user's movements to the virtual avatar [40]. The dance training system guides the user through a series of exercises, and the user is assigned an overall score at the end of the exercises. The system creates the score

by analyzing the musical motion features (MMF) for guidance and rhythm and laban movement analysis (LMA) for movement style.

Dancing Salsa with Machines: Filling the Gap of Dancing Learning Solutions

Dancing Coach (DC) is a training system that helps a person practice the basic salsa dancing steps [38]. The system uses Kinect V2 to track the user's facial expression and body movements. There is a tutorial mode which allows the user to practice the dance steps without any feedback. While the user practices a sequence of 8 basic salsa dancing steps, the training system will indicate which foot to move. The practice mode allows the user to get feedback and enable audio beat support. The feedback evaluates the user on the following categories: smiling, looking straight, avoiding stiffness, and dancing on the beat.

An Approach to Ballet Dance Training through MS Kinect and Visualization in a CAVE Virtual Reality Environment

A cave automatic virtual environment (CAVE) is created to instruct and evaluate ballet dance training [27]. The system uses a Microsoft (MS) Kinect camera system to capture the user's dance movements. From the captured dance sequence, the user's dance pose is compared to a gesture database which contains the teacher's dance movements. The training system then evaluates the ballet movements and creates a score graph based on how closely the student's performance matches the teacher's performances. The feedback mode allows the user to replay the performance where the teacher and student's movements are side by side or overlapped.

3.3.3 Dance Imitation

Towards Bi-directional Dancing Interaction

A virtual rap dancer is created as an embodied conversational agent (ECA) to interact with the human [36]. The virtual dance system includes a camera, dance pad and beat detection to create the virtual dance partner. In order to animate the virtual character, movement is chosen from a database that contains information about the frame animation and human pose. This virtual performer has the following states: bored, invite and dance. The interaction model will have the dance partner mimic a bored-behavior in the bored state, but the invite state will have the virtual partner gesture to come dance. The dance state has two options: following or leading. The following phase will mimic the user while the leading phase will vary the movements.

Everybody Dance Now

The model uses motion transfer to transfer the performance of an amateur dancer to a performer, such as Bruno Mars. In order to obtain human motion transfer, a pose detector identifies the amateur dancer's poses and applies the learned pose-to-appearance mapping to generate the performer [8]. A generative adversarial network (GAN) trains and evaluates on a dataset that contains five long single-dancer videos and a large collection of short YouTube videos. The system architecture first applies pose detection on the input video then applies global pose normalization. Using the normalized skeleton, the model maps the poses to the performer.

3.3.4 Music-oriented Dance Partner

Music-oriented Dance Video Synthesis with Pose Perceptual Loss

By incorporating automatic music video generation, the model creates a realistic video to conform to any song's beats and rhymes. First the music is used for the human skeleton sequence synthesis, and then a synthesized video is generated by pose-to-appearance mapping [37]. Using pose perceptual loss, the model can train on imperfect human poses to create realistic dance skeleton sequences. The framework uses a dataset that contains paired music and skeleton sequences. The following datasets were used: K-pop dataset, Let's Dance Dataset and FMA.

Interacting with a Virtual Rap Dancer

Using audio and video signals, a virtual rap performer is created to dance to the music and motion beats [35]. The dancer is a VRML avatar that uses a database of movements from 9 different videos of rap songs. The virtual dance trains on the database to mimic the dance poses. During the song, the virtual dance partner will use a beat prediction algorithm to choose their next move based on the beat of the music.

3.3.5 Special Effects for Dance

Interactive Augmented Reality for Dance

Applied to live augmented reality productions, ViFlow is a fully interactive and real-time system for dance performances. Compared to creating a dance production and incorporating digital projected spaces, ViFlow is a simple and cost effective system [5]. This system creates a silhouette by tracking the infrared radiation (IR) reflection

of the dance performer. Using the silhouette, the system determines the body joints to apply the virtual effects and create an interactive performance. The architecture requires a camera that can detect light in the infrared spectrum and infrared light emitters. The software is comprised of tracking and rending modules in order to create the dynamically generated backdrop image.

3.3.6 Overview

These interactive dance partners often required motion capture data or a physical robot to be created which is usually expensive. The dance training systems will also use a motion capture system or virtual reality technologies to record, analyze and provide feedback on the user's dance movements. Some of the dance training systems will provide visual feedback during the dance practices, such as highlighting the dance steps or dance positions. There are a couple of systems that incorporate virtual reality by creating a virtual dance teacher and recreating the user as the student.

Human motion transfer is applied to generate a dance performer, such as Bruno Mars, that mimics the user's dance moves. Another dance imitation system uses a dance pad to create a virtual rap dancer that mimics the user's dance movements. By using the beat of the music, virtual dance partners are created to dance to the song. Based on the dancer's movements, there is an interactive augmented reality system that creates special effects during a performance.

The most similar approach to my work is the creation of a virtual salsa dance partner which uses a motion capture or virtual reality system [22] [31]. The main difference is my approach will only require the model to be trained off of YouTube videos which are free and easily obtainable.

CHAPTER 4

SWING DATASET

In terms of difficulty level, the challenge with choosing advanced dancers is that the movements will not always be a consistent dance sequence since some dancers will add additional moves or deviate from a standard sequence. The advance dancers will also move at faster paces. Intermediate dance moves usually have a combination of basic dance steps while adding technique. This mixture of dance moves could also be challenging to identify since consistent dance moves are not guaranteed. Beginner dance moves were chosen because there will usually be a finite amount of dance positions that are clear and easy to follow.

Dance datasets, YouTube, dance competitions and dance instructor websites were considered when collecting the videos. The challenge with dance datasets was each dataset varied in quality, size and dance types. Dance instructor websites, such as BallroomDancers.com, had better quality but very few videos to offer. Dance competitions provide a variety of dance partners, but often times the video background can be cluttered with objects, orchestras, or multiple partner dances. Typically, the dance competitions were international or advanced level which creates inconsistent dance sequences.

Since YouTube offers a free collection of videos, YouTube was the main resource for dance videos. YouTube also provides many dance channels that have slow and easy dance instructional videos for beginners. Another advantage is that these instructional YouTube videos contain multiple clips of the dancers performing a dance move or sequence. By creating a larger dataset to train a model, these swing dance videos

would also often contain different angles where the lead and follow would switch positions so that they alternate dance partners in front of the camera.

To avoid OpenPose misidentifying the main dance couple in a video frame, YouTube videos were selected if there was a single dance couple with a clear background. The dancers must be wearing plain clothes, and the videos must have good resolution where each dance partner can be visually identified. The following types of swing dance were chosen because they are the most popular in the United States and around the world: East Coast, West Coast and Lindy Hop [43]. There is also a large amount of beginner tutorials available on YouTube for the 3 different dance styles.

East Coast Swing is a common type of swing dance that mainly uses 6-count patterns, but there are a few 8-count patterns. This swing dance is a more of a rotational dance and is typically fast, upbeat, energetic and bouncy [42]. The main dance moves are rock step and triple steps. The dance patterns often end in a rock step and can be performed in single, double, and triple rhythms. Figure 4.1 shows an overview of the East Coast Swing videos.

West Coast Swing is another beginner friendly type of swing dance. This dance style has the same 6-count patterns as East Coast Swing, but the speed and the music is different. This dance can be applied to a wide variety of music genres, such as slow and fast blues, pop or country [42]. This dance style focuses on a smooth dance with a strong partner connection. West Coast Swing is a slotted dance which means the dancers stay in a rectangular area. One basic dance sequence is a walk, walk, triple step, triple step. Figure 4.2 shows an overview of the West Coast Swing videos.

Lindy Hop is another one of the most common swing dance styles to learn as a beginner. This dance style typically includes 6-count and 8-count patterns with flips, kicks and swing-out moves [43]. Some of the basic moves include rock step and triple

steps. Lindy Hop involves a lot of movement across the dance floor and typically has a fast rhythm. Figure 4.3 shows an overview of the Lindy Hop videos.

Overall these 3 types of swing dance share some similar basic dance steps, but also present different dance styles. East Coast, West Coast and Lindy Hop videos were included in the dataset to create a variety of dance movements for the model to train on. My dataset contributes a swing dance dataset that contains 712 videos clips from 60 videos. There are 17 different YouTube channels that contain beginner instructional swing dance videos. There are a few videos that include footwork only or a little improvisation, but the majority of videos focus on basic dance moves with a partner. Table 4.1 gives an overview of the swing dataset.

Table 4.1: Swing Dataset Overview

Swing Dance Type	YouTube Channels	Videos	Video Clips	Video Frames
West Coast	1	1	214	92,119
East Coast	9	27	206	74,046
Lindy Hop	7	32	292	121,481
Total	17	60	712	287,646

Table 4.2 gives an overview of the amount of storage for the swing dataset. The number of videos represents the original 60 YouTube videos. The number of video clips represents the 712 video clips cut directly from the original YouTube videos. The videos are saved as MP4 format. The number of video frames account for the video frames after OpenPose was run on the original YouTube videos. The video frames are saved as a PNG format. The number of skeleton keypoints represents the recorded keypoints that OpenPose identified on each video frame. After OpenPose processes the video frames, these skeleton keypoints are saved in a JSON file. The

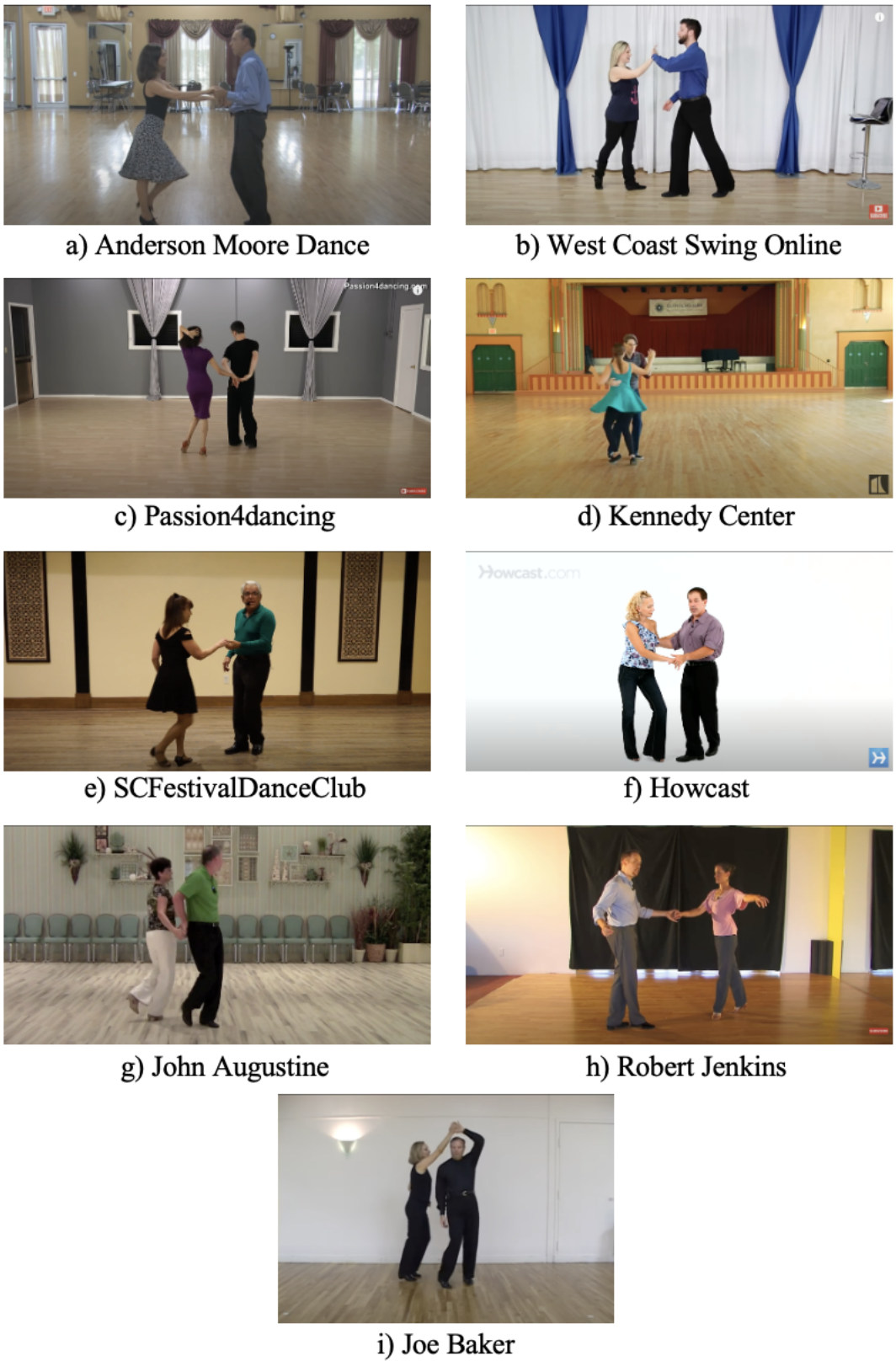
skeleton keypoints are then normalized and saved into a NumPy file which represents the normalized skeleton keypoints.

Table 4.2: Dataset Storage Overview

Swing Dance Type	Videos	Video Clips	Video Frames	Skeleton Keypoints	Normalized Skeleton Keypoints
West Coast	1.02 GB	2.26 GB	935.0 MB	152.5 MB	121.3 MB
East Coast	1.87 GB	2.19 GB	723.1 MB	116.1 MB	96.0 MB
Lindy Hop	1.17 GB	3.43 GB	1.2 GB	191.3 MB	156.4 MB
Total	4.06 GB	7.88 GB	2.86 GB	459.9 MB	373.7 MB

The instructional videos would often pause in between dance movements, show different camera angles, or film close ups when teaching a beginner dance sequence. This swing dataset crops the videos to only contain the portions of the instructional video that show a continuous dance sequence with no pauses or interruptions. The videos always contain 2 dancers which are a lead and follow, and the dancers wear plain clothes. Usually the lead is male and the follow is female except for one video within West Coast swing dance where the lead and follow are both female dancers.

The YouTube videos are recorded with a static camera position and in an empty room. There are no extra people in the background to minimize the problem of misidentifying a person as the main dance couple. The length of each video clip ranges from approximately 4 seconds to approximately 1 minute. The videos chosen are typically high quality with 1280p x 720p resolution. Out of the 60 videos, 4 videos have 640p x 360p resolution and 1 video has 480p x 360p resolution. Even though some lower resolution videos are included in this dataset, the dance partners and movements are clearly visible.



a) Anderson Moore Dance

b) West Coast Swing Online

c) Passion4dancing

d) Kennedy Center

e) SCFestivalDanceClub

f) Howcast

g) John Augustine

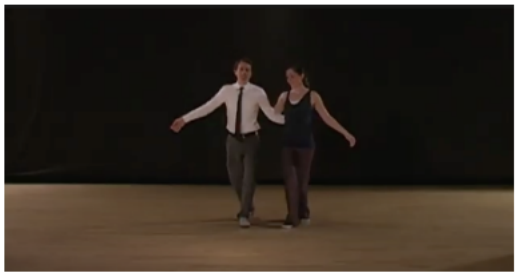
h) Robert Jenkins

i) Joe Baker

Figure 4.1: East Coast Swing

Figure 4.2: West Coast Swing

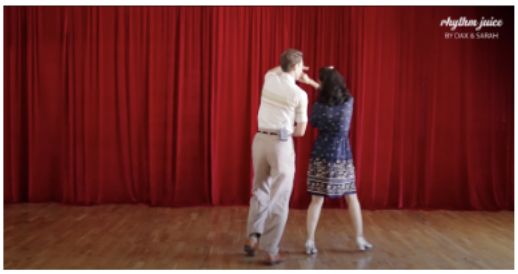
a) SwingStepTV



b) Joe DeMers

c) Rhythm Dance City

d) Lindy Hop St. Louis



e) Rhythmjuice

f) Sylvie Rene

g) iLindy

Figure 4.3: Lindy Hop

CHAPTER 5

METHODS

In order to preprocess the YouTube videos, iMovie is used to cut and create the 712 video clips from the 17 different beginner tutorial swing dance YouTube channels. Since iMovie saves the video clips in a video format that does not work with OpenPose, FFmpeg is used to convert the videos to a lower compressed MP4 format while keeping the best quality. For each video frame, OpenPose identifies the skeletons and formatted the video to highlight the skeletons with a black background. This program also converts the videos into skeleton keypoints which are saved in a JSON file. Each JSON file represents a frame and stores the keypoints for each person in an array.

After converting the video frames to skeleton keypoints, the skeleton keypoints are normalized. Each frame is normalized by selecting the main dance couple within the frame. There are a couple of videos where OpenPose would occasionally misidentify an object for a person, such as a sewing display dress form. To account for this scenario, the main dance couple is selected by calculating the height of all the skeletons in a frame and selecting the largest skeletons. Typically, the dancers are in front of the background objects which guarantees that the main dance couple would have the largest skeleton sizes in a single frame.

Once the main dancers are identified in the video frame, the skeletons are normalized by finding the centroid from the pair of skeletons and removing the value from each skeleton keypoints. The skeleton keypoints are further normalized by finding the height from the pair of skeletons and dividing each dancer's skeleton keypoints by

the calculated average height. For each video frame, the normalized keypoints of the main dancers are saved to a numpy file.

OpenPose will attach a confidence score to each body part which will rate how likely the body part is correctly identified. Since OpenPose often misidentified body parts, the body parts with low confidence scores will be imputed. For body joints that had a confidence score of 0, NaN values are inserted for the x and y coordinates. These formatted and normalized keypoints are then passed to Scikit Learn's KNNImputer which will use k-Nearest Neighbors to guess where the missing body parts would be located.

Before I used Scikit Learn's KNNImputer, the KNNImputer was tested for multiple scenarios by hiding multiple joints or different body parts. I created a dataset that contained only normalized skeletons with no missing joints. A train dataset was created by taking 90% of the dataset, and a test dataset was created by taking the remaining 10% of the dataset.

A prediction dataset was created by taking the test dataset and randomly hiding specific joints. The missing joints were created by randomly inserting NaN values for each skeleton keypoints. The KNNImputer learned to impute the missing values on the train dataset. Using the prediction dataset, the model predicted the missing joints on unseen data. To verify the KNNImputer predicted the hidden joints correctly, I calculate the average distance between the real and predicted joints.

After verifying the KNNImputer will predict reasonable estimates for the missing joints, the normalized numpy files are then combined into one Pandas dataframe which is saved to a CSV file. The pandas dataframe allows the keypoints to be potentially split by different attributes. This dataframe contains the following features: skeleton id, YouTube channel, video name, partner type, frame file path, and the

list of keypoints for a skeleton. The partner type identifies whether the skeleton is the lead or follow within each frame. Table 5.1 gives an overview of the normalized skeleton keypoints that are saved in the dataframe.

Table 5.1: Normalized Skeleton Keypoints Dataframe

Id	YouTube Channels	Video Name	Partner Type	Frame Path	x0	y0	c0	...	x24	y24	c24
1	Howcast	vid03	1	path1	0.14	-0.47	0.89	...			
2	iLindy	vid17	1	path2	0.33	0.79	0.19	...			
3	iLindy	vid25	2	path3	0.55	-0.61	0.64	...			

OpenPose typically records two skeleton keypoints when two performers held different dance positions. Sometimes, two skeletons overlap in a frame, such as the lead performer is behind the follow performer for a dance move or vice versa. If the dancers overlap, then OpenPose only records one of the dancer's keypoints. When OpenPose identifies only one skeleton in the frame, the skeleton keypoints are duplicated. If the two skeletons overlap, then their body keypoints should be similar.

Before the missing joints are imputed, the list of normalized skeleton keypoints was horizontally flipped to create more data and saved to a Pandas dataframe. Figure 5.1 shows the original keypoints and flipped keypoints for a skeleton. KNNImputer assigned values to the missing joints in the dataset. The imputed dataset is then passed to 3 different models: linear regressor, KNeigbhors regressor and extra trees regressor.

Each model will predict the opposing partner's dance moves based on the input of the current dancer in a frame. The imputed dataset is split into train and test datasets using a 10% split based on the number of videos. A random seed was set so that the split will be consistent across the models. Each model learns on their respective train dataset and creates a prediction dataset by fitting the model on the

test dataset. The models are then evaluated by computing the following metrics: average Euclidean distance between keypoints and standard deviation. Figure 5.2 gives a general overview of the process to predict one skeleton based off the the input pose from another skeleton.

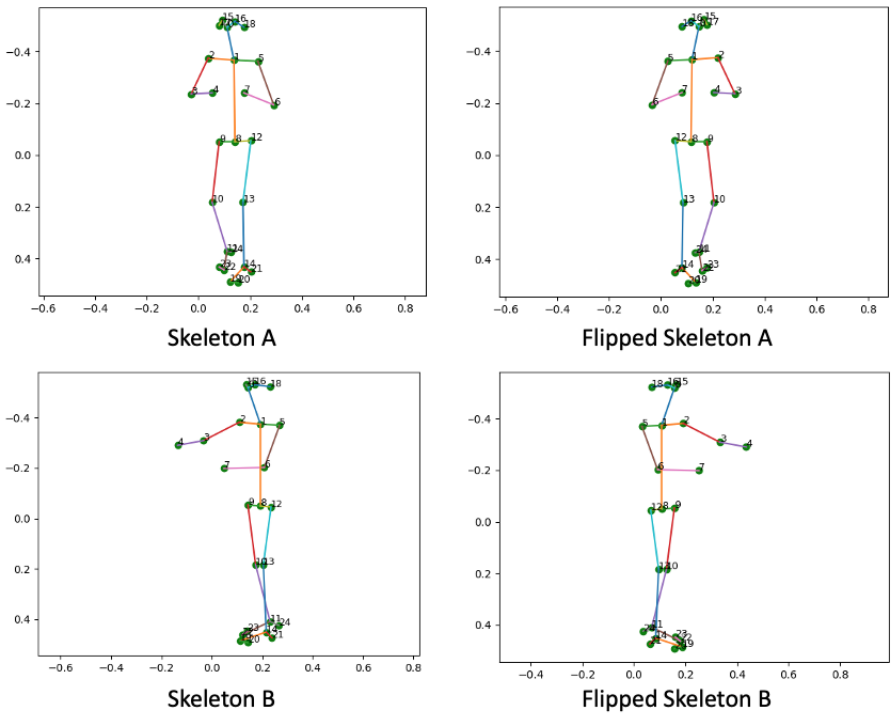


Figure 5.1: Horizontally Flipped Skeletons

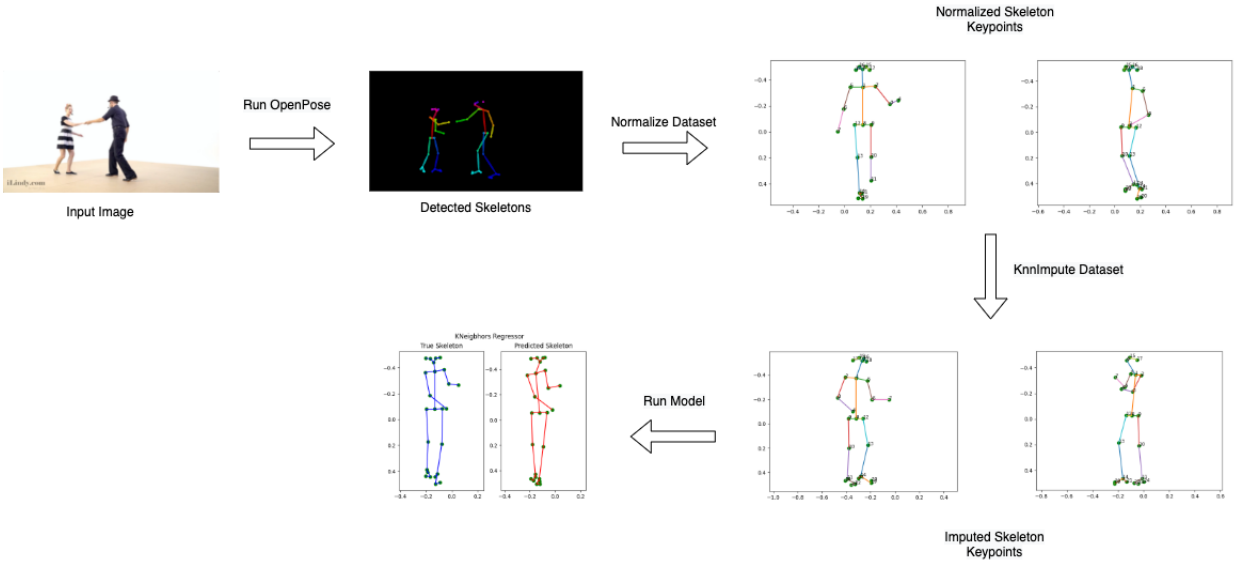


Figure 5.2: System Architecture

CHAPTER 6

RESULTS AND DISCUSSIONS

Table 6.1 gives an overview of the metrics when using the KNNImputer to predict the missing joints in the skeleton keypoints. The results show that KNNImputer works well when there are only a few random joints missing instead of an entire body part. If only 1 random joint is missing for each skeleton, then there are more skeleton keypoints to predict from. When a body part is hidden, 3 keypoints will always be missing for each skeleton's keypoints.

Table 6.1: Average Distance Between Real and Predicted Joints for Hidden Joints

Hidden	Average Distance Between Real and Predicted Joints
1 Random Joint	0.0090
2 Random Joints	0.0088
3 Random Joints	0.0089
Arm	0.0137
Foot	0.0109
Leg	0.0086

The lack of body keypoints could be the reason why the K-Nearest Neighbors algorithm performs worse on hidden body parts instead of a few hidden joints. The KNNImputer will not train on the location of the hidden body part and will predict the missing body part based on the surrounding joints. This effect could cause the KNNImputer to inaccurately predict the position of a body part more often.

The average distance between the real and predicted joints are measured relative to the height of the skeleton which was normalized to 1. I was aiming to get less than

10% error and the highest average distance was 1.37%. Figure 6.1 shows qualitatively that the KNNImputer's predicted skeleton keypoints matches closely to the original skeleton keypoints. Since the videos typically are only missing a few joints for each skeleton, these metrics verify that the KNNImputer will work for imputing the missing joints.

Table 6.2 contains the mapping of the body keypoints that represent the hidden body parts on a skeleton. Figure 6.1 visually shows how well KNNImpute predicts the missing body joints. The displayed tests show that when an entire arm or a leg is hidden, KNNImpute can predict the location of the missing body part. Both skeleton A and skeleton B have similar body joint positions compared to the original skeleton.

Table 6.2: Mapping of Body Parts to Skeleton Keypoints

Hidden Body Parts	Hidden Joints
Arm	2, 3, 4
Leg	12, 13, 14
Foot	22, 23, 24

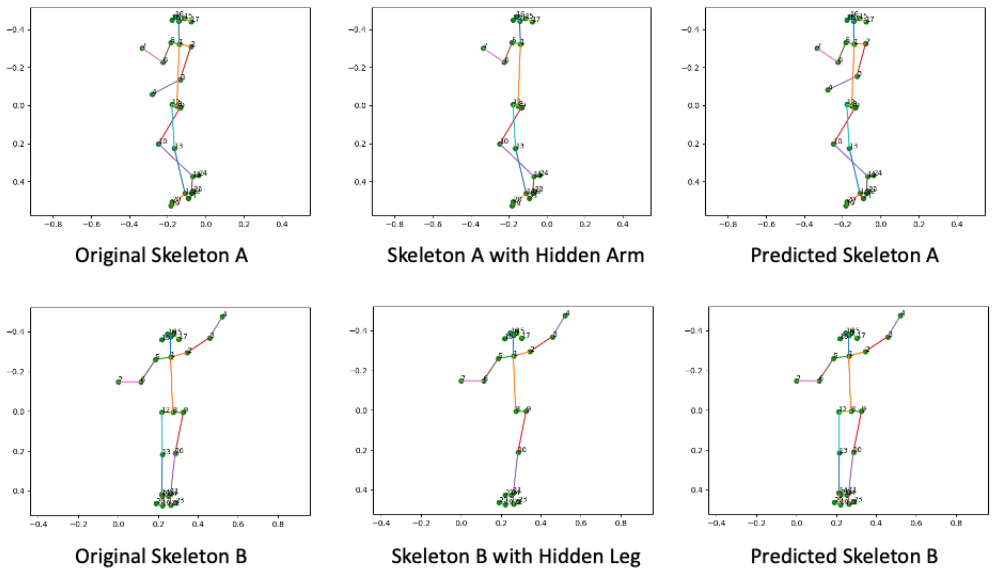


Figure 6.1: Verifying KNNImpute by Hiding Body Parts

Table 6.3 gives an overview of the metrics for the 3 different regression models. Based on the average euclidean distance between keypoints and standard deviation, the model performance ranks in the following ascending order: linear regressor, K-Nearest Neighbors regressor, and Extra Trees regressor.

The linear regression model performs the worst because the model assumes there is a linear relationship between the skeleton keypoints. The dancer can spin, turn, and cross their legs which creates a wide range of keypoint locations. The distribution and position of body joints is not as suitable to create a linear relationship between the real and predicted joints which could create the high average euclidean distance between the keypoints. The linear regression model is sensitive to outliers which could also explain the higher standard deviation.

K-Nearest Neighbors regressor performs well because the algorithm predicts the keypoints based on the keypoints in close proximity. This approach works well since a person's body parts typically stay in the same general region. For example, a person's head will anatomically never be below the legs. The K-Nearest Neighbors regressor still produces some error which could be caused by the changes in orientation. If the dancer is often turning and changing angles in the video, then the algorithm might group the motions together. This could explain why the arms are sometimes plotted in the opposite direction or why the legs are sometimes predicted too close or far apart. The standard deviation could be higher than the Extra Trees standard deviation because the K-Nearest Neighbors regressor is sensitive to outliers because the algorithm groups the keypoints based on distance.

Extra trees regressor performs the best out of the 3 models which could be caused by the algorithm creating a large number of decision trees. By calculating the mean from 100 decision trees, the prediction could be more accurate. The method randomly splits and samples features of the decision tree which could help the model generalize

the joint predictions. This effect could lead to the lowest average euclidean distance between keypoints. The randomization of keypoints may have helped the model learn more and different attributes which created a lower standard deviation.

Table 6.3: Model Metrics

Regression Model	Average Euclidean Distance Between Keypoints	Standard Deviation
Linear	0.0847	0.0782
K-Nearest Neighbors	0.0511	0.0580
Extra Trees	0.0505	0.0521

Qualitative Results

Figure 6.2 shows an example of one dance couple selected from a video frame. Once the dance couple is split into individual skeletons, one skeleton is selected to be the input for the 3 models. In this scenario, the lead skeleton is the input. The other skeleton is selected to be the true skeleton which is the actual skeleton that the models are trying to predict. In this case, the true skeleton is the follow from the dance couple.

Figure 6.3 shows the model predictions for dance couple 1. Compared to the true skeleton in figure 6.2, the K-Nearest Neighbors regressor predicts the head towards the right instead of the left. The predicted skeleton has no right arm and the knees are inverted. The footwork is somewhat similar to the true skeleton but needs to slightly raise on of the legs.

The linear regressor is the worst at matching the the true skeleton. The skeleton is displayed as if the camera is capturing the dancer from the side. The predicted skeleton is facing towards the right when the true skeleton has the neck pointed to the left. The footwork does not match the actual skeleton.

In this example, the Extra Trees regressor performs the best at matching the general structure of the true skeleton. Similar to the other model predictions, the neck on the predicted skeleton is pointed in the wrong direction, but the skeleton has a bend in both arms. The model predicts the arms lower than the true skeleton, but gets the overall shape. Similar to the actual skeleton, the predicted legs are spaced out. Only one foot needs to be slightly raised.

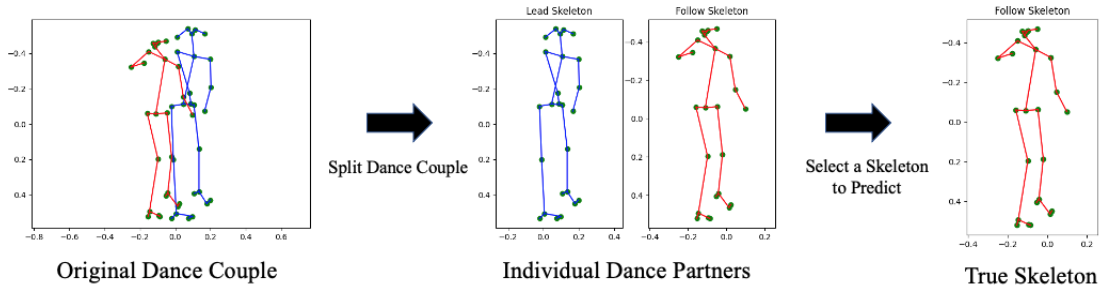


Figure 6.2: Selected Skeleton to Predict from Dance Couple 1

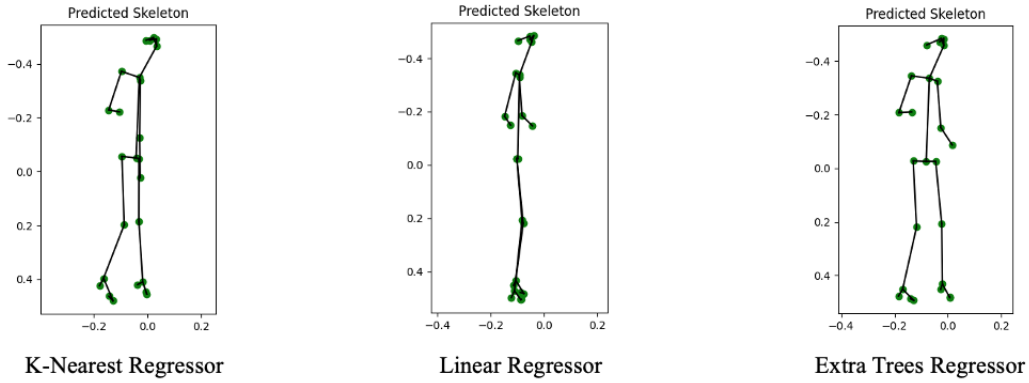


Figure 6.3: Model Predictions for Dance Couple 1

Figure 6.4 shows a different dance couple selected. The lead skeleton is the input to the 3 different models and the follow skeleton is the actual skeleton that the models attempt to predict. Figure 6.5 shows the model predictions for dance couple 2. In this scenario, the K-Nearest Neighbors regressor makes the best prediction. The overall structure of the skeleton and position mostly matches the true skeleton. Compared

to the actual skeleton, the predicted arms are a little bent inwards and the head is more spread out.

Out of the 3 models, linear regressor performs the worst. Resembling a straight line, the predicted skeleton has all the body parts compressed and the predicted skeleton is facing the opposite direction of the true skeleton. The model could be predicting that the performer is dancing with their side facing to the camera.

Matching the direction of the true skeleton, the Extra Trees regressor has the predicted skeleton facing towards the right. The legs are too close together and the arms are bent in the same direction instead of one arm up and the other arm down. This model could also be predicting the camera angle is pointed at the side of the dancer because the hips and knees overlap.

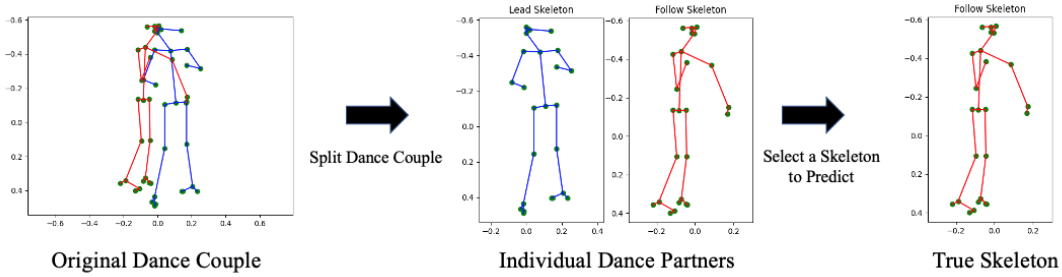


Figure 6.4: Selected Skeleton to Predict from Dance Couple 2

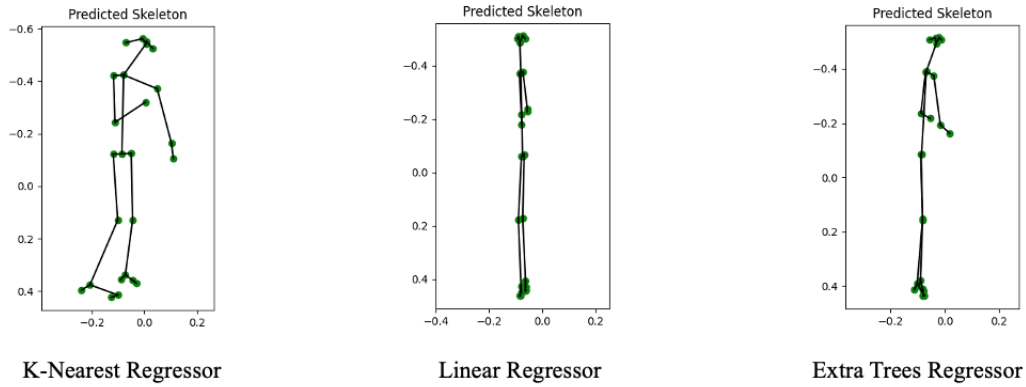


Figure 6.5: Model Predictions for Dance Couple 2

Figure 6.6 shows a third dance couple selected. The lead skeleton is the input to the 3 different models and the follow skeleton is the actual skeleton that the models attempt to predict. Figure 6.7 shows the K-Nearest Neighbors regressor matches the general shape of the true skeleton, but the predicted skeleton is too stretched. Compared to the true skeleton, the predicted arms and legs are to far apart and the head is positioned too far forward.

The linear regressor is close to predicting the true skeleton based on the overall shape. For the predicted skeleton, the arms are too far apart and the legs are slightly crossed. The model could be predicting that the dancer's chest is facing more towards the camera while the feet are slightly crossed. The head is positioned correctly and the spacing between the legs is close to the true skeleton.

In this example, the Extra Trees regressor performs the best because the predicted skeleton is the closest to matching the overall structure and spacing of the body parts. For the predicted skeleton, the arms and the legs need to be spaced a little more, but the general position is similar to the actual skeleton. The predicted head is in the right direction and the predicted footwork closely matches the true skeleton's footwork.

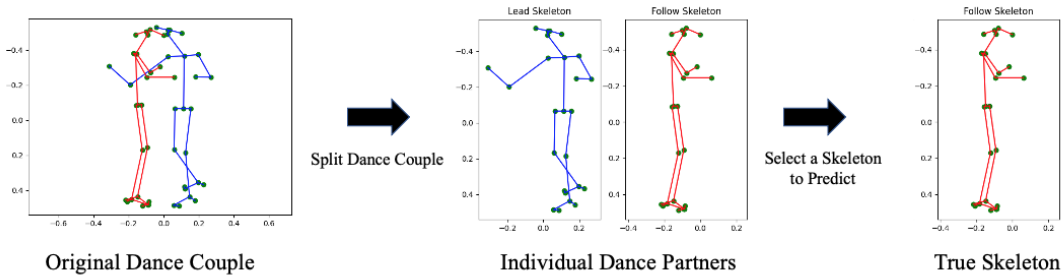


Figure 6.6: Selected Skeleton to Predict from Dance Couple 3

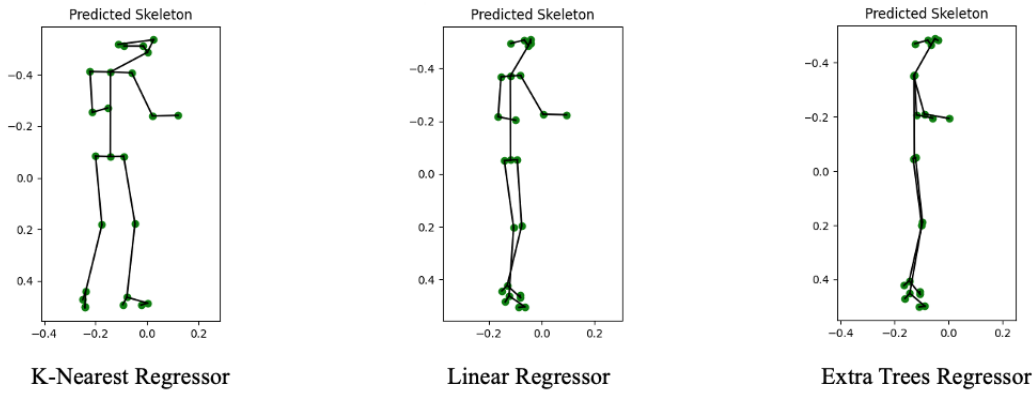


Figure 6.7: Model Predictions for Dance Couple 3

CHAPTER 7

CONCLUSION

My thesis created a large swing dance dataset that contains 712 video clips from 17 different YouTube channels. This dataset highlights 3 different types of swing dances that are most common among beginner swing dancers. The swing dataset also contains a variety of dancers and backgrounds. After preprocessing the videos with OpenPose and normalizing the skeleton keypoints, KNNImputer was verified and used to impute the missing skeleton keypoints. Based on the average distance between real and predicted joints, the KNNImputer had less than a 10% error.

The following models trained on the normalized dataset: linear regressor, K-Nearest Neighbors regressor and Extra Trees regressor. Based on the average euclidean distance between keypoints, the linear regressor preformed the worst and the extra trees regressor performed the best.

By combining the swing dance datasets and models, my approach can predict the opposing partner's dance movements based on the input of the current dancer in the same frame. My method is cost effective because the swing dataset only relies on YouTube videos instead of a motion capture system or virtual reality environment. In addition, my work presents a survey for existing dance datasets and dance applications that combine computer vision and dance.

CHAPTER 8

FUTURE WORK

Instead of predicting the dancer partner's movement based off the dancer in the current frame, the system could predict the dance partner's next couple of dance movements based off the dancer in the current frame. Using the swing dance dataset, a training program could be created to teach a person how to swing dance and an avatar could be created to help guide the user. The user could record a video of themselves dancing and upload the video to a program that scores the performance. The dance training system could display the virtual dance partner on the original video as a visual feedback.

Another potential direction is to create a virtual dance partner where the user can record themselves through their laptop camera and the virtual dance partner would display on their screen in real-time. Incorporating depth, the system could also create an interactive virtual dance partner in 3D [31] [22]. OpenPose can estimate 3D skeleton keypoints, so the user could be captured in real-time and the virtual dance partner could be recreated.

One direction is to expand the dataset to record at different angles to allow the user to practice dancing with a virtual at different angles. The dataset could also be annotated with the dance moves and dance sequences for classification and recognition problems. For each video frame in the dataset, the lead and follow could be labeled in order to keep track if the dancers switch places while turning. The dataset could be expanded to also include different backgrounds that include audiences, spectators, or a lot of objects like instruments. The dataset could include more levels of swing

dance, such as advanced dance moves, international swing dance or swing dance competitions. This approach could also be applied to different types of dance, such as ballet or tango.

One current limitation is that OpenPose often inaccurately estimated some of the joints in each skeleton or misidentified objects as people. One solution is to create a better preprocessing step that identifies the main dance couple in a frame instead of finding the largest skeletons based on height. Another limitation was the models did not always predict an exact matching skeleton and would sometimes predict a skeleton that was not anatomically correct, such as the eyes would be spread too far apart. One way to solve this problem would be to calculate if the body parts are reasonably positioned and if the skeleton joints are misplaced then impute those body joints.

BIBLIOGRAPHY

[1] B. Allain, J.-S. Franco, and E. Boyer. An efficient volumetric framework for shape tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 268–276, 2015.

[2] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[3] P. Aznar. What is the difference between extra trees and random forest?, Jun 2020.

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] T. Brockhoeft, J. Petuch, J. Bach, E. Djerekarov, M. Ackerman, and G. Tyson. Interactive augmented reality for dance. In *Proceedings of the Seventh International Conference on Computational Creativity*, pages 396–403, 2016.

[6] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018.

[7] D. Castro, S. Hickson, P. Sangkloy, B. Mittal, S. Dai, J. Hays, and I. A. Essa. Let's dance: Learning from online dance videos. *CoRR*, abs/1801.07388, 2018.

[8] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros. Everybody dance now. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5933–5942, 2019.

[9] J. C. Chan, H. Leung, J. K. Tang, and T. Komura. A virtual reality dance training system using motion capture technology. *IEEE transactions on learning technologies*, 4(2):187–195, 2010.

[10] S. H. Chavoshi, B. De Baets, T. Neutens, G. De Tré, and N. Van de Weghe. Exploring dance movement data using sequence alignment methods. *PLOS ONE*, 10(7), 2015.

[11] T. L. Chen, T. Bhattacharjee, J. M. Beer, L. H. Ting, M. E. Hackney, W. A. Rogers, and C. C. Kemp. Older adults' acceptance of a robot for partner dance-based exercise. *PLOS ONE*, 12(10), 2017.

[12] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[13] S. Dewan, S. Agarwal, and N. Singh. Automatic labanotation generation, semi-automatic semantic annotation and retrieval of recorded videos. In M. Dobreva, A. Hinze, and M. Žumer, editors, *Maturity and Innovation in Digital Libraries*, pages 55–60, Cham, 2018. Springer International Publishing.

[14] Efros, Berg, Mori, and Malik. Recognizing action at a distance. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 726–733 vol.2, 2003.

[15] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[16] E. Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[17] J. Friedman, T. Hastie, R. Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[18] Genesis. Pros and cons of k-nearest neighbors, Sep 2018.

[19] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[20] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.

[21] D. Hendry, K. Chai, A. Campbell, L. Hopper, P. O'Sullivan, and L. Straker. Development of a human activity recognition system for ballet tasks. *Sports Medicine - Open*, 6, 02 2020.

[22] E. S. L. Ho, J. C. P. Chan, T. Komura, and H. Leung. Interactive partner control in close interactions for real-time applications. *ACM Trans. Multimedia Comput. Commun. Appl.*, 9(3), July 2013.

[23] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset, 2017.

[24] Y. Kim and D. Kim. Real-time dance evaluation by markerless human pose estimation. *Multimedia Tools and Applications*, 77(23):31199–31220, 2018.

[25] K. Kosuge, T. Takeda, Y. Hirata, M. Endo, M. Nomura, K. Sakai, M. Koizumi, and T. Oconogi. Partner ballroom dance robot-pbdr. *SICE Journal of Control, Measurement, and System Integration*, 1(1):74–80, 2008.

[26] M. Kyan, G. Sun, H. Li, L. Zhong, P. Muneesawang, N. Dong, B. Elder, and L. Guan. An approach to ballet dance training through ms kinect and visualization in a cave virtual reality environment. *ACM Trans. Intell. Syst. Technol.*, 6(2), Mar. 2015.

[27] M. Kyan, G. Sun, H. Li, L. Zhong, P. Muneesawang, N. Dong, B. Elder, and L. Guan. An approach to ballet dance training through ms kinect and visualization in a cave virtual reality environment. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(2):1–37, 2015.

[28] U. Marchand and G. Peeters. The Extended Ballroom Dataset, Aug. 2016. Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conf.. 2016.

[29] H. Matsuyama, K. Hiroi, K. Kaji, T. Yonezawa, and N. Kawaguchi. Ballroom dance step type recognition by random forest using video and wearable sensor. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, UbiComp/ISWC '19 Adjunct, page 774–780, New York, NY, USA, 2019. Association for Computing Machinery.

[30] H. Matsuyama, K. Hiroi, K. Kaji, T. Yonezawa, and N. Kawaguchi. *A Basic Study on Ballroom Dance Figure Classification with LSTM Using Multi-modal Sensor*, pages 209–226. Springer Singapore, Singapore, 2021.

[31] C. Mousas. Performance-driven dance motion control of a virtual partner character. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 57–64, 2018.

[32] T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang, and C. Yang. The progress of human pose estimation: A survey and taxonomy of models applied in 2d human pose estimation. *IEEE Access*, 8:133330–133348, 2020.

[33] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *CoRR*, abs/1603.06937, 2016.

[34] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. *CoRR*, abs/1511.06645, 2015.

[35] D. Reidsma, A. Nijholt, R. Rienks, and H. Hondorp. Interacting with a virtual rap dancer. In *International Conference on Intelligent Technologies for Interactive Entertainment*, pages 134–143. Springer, 2005.

[36] D. Reidsma, H. Van Welbergen, R. Poppe, P. Bos, and A. Nijholt. Towards bi-directional dancing interaction. In *International Conference on Entertainment Computing*, pages 1–12. Springer, 2006.

[37] X. Ren, H. Li, Z. Huang, and Q. Chen. Music-oriented dance video synthesis with pose perceptual loss. *CoRR*, abs/1912.06606, 2019.

[38] G. Romano, J. Schneider, and H. Drachsler. Dancing salsa with machines—filling the gap of dancing learning solutions. *Sensors*, 19(17):3661, 2019.

[39] A. R. Rout. Ml - advantages and disadvantages of linear regression, Jun 2020.

[40] S. Senecal, N. A. Nijdam, A. Aristidou, and N. Magnenat-Thalmann. Salsa dance learning evaluation and motion analysis in gamified virtual reality environment. *Multimedia Tools and Applications*, 79(33):24621–24643, 2020.

[41] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012.

[42] StaffLogin. East coast swing vs. west coast swing - what is the difference?, Mar 2019.

[43] StaffLogin. Types of swing dance - ecs, wcs, lindy, jive & more, Apr 2019.

[44] E. Stavrakis, Y. Chrysanthou, and A. Aristidou. Dance motion capture database, 2021.

[45] F. Tanaka, J. R. Movellan, B. Fortenberry, and K. Aisaka. Daily hri evaluation at a classroom environment: Reports from dance interaction experiments. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, HRI '06, page 3–9, New York, NY, USA, 2006. Association for Computing Machinery.

[46] T. Tang, J. Jia, and H. Mao. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1598–1606. ACM, 2018.

[47] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013.

[48] S. Tsuchida, S. Fukayama, M. Hamasaki, and M. Goto. Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *ISMIR*, pages 501–510, 2019.

[49] A. Vögele and B. Krüger. Hdm12 dance - documentation on a data base of tango motion capture. Technical Report CG-2016-1, University of Bonn, Sept. 2016. ISSN 1610-8892.

[50] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016.