INTRODUCTION

1.1 **Human Motion Prediction**

Humans are blessed with a remarkable ability to make accurate short-term predictions about their surroundings based on past observations (Gui et al., 2018). Crossing a crowded street is one among many other tasks that would be very challenging without our capacity of understanding human movements and anticipating their most likely actions in the near future (Martinez et al., 2017). Given a set of 3D poses or skeletons, the goal of human motion prediction is to forecast the most likely future frames of human motion based on the given sequence as illustrated in Figure 1.1 (Tang et al., 2018; Martinez et al., 2017).

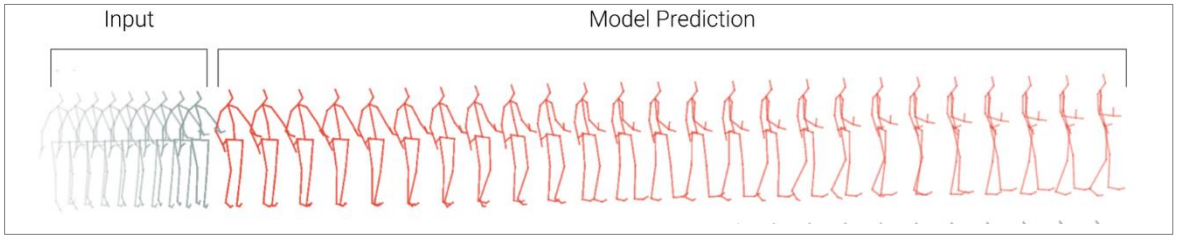

Figure Error! No text of specified style in document.**.1: Motion prediction task. The gray colored sequence is the input sequence and the red colored sequence is the output sequence or the prediction (Martinez et al., 2017).**

Human motion prediction is essential for achieving the goal of robotic intelligence where robots are supposed not only to have a notion of human motion but also to be able to predict their movements, resulting in a seamless interaction between humans and machines (Tang et al., 2018). Many situations involve human-robot interaction such as handshaking during socialization or handing tools to a surgeon during an operation. For the interaction to be successful, the robot is supposed to recognize and forecast limbs' pose and position precisely so that it can provide a rapid and proper response (Tang et al., 2018). Human motion forecasting proved to be important for plenty of tasks including action detection, action recognition and action analysis in computer vision, body pose estimation as well as motion synthesis in computer graphics, virtual and augmented reality, etc. (Pavllo at al.,

2018). Humans by nature are very flexible and can perform complex movements that are subject not only to the physical laws but also to the intentions of the moving person. Thus, human motion is inherently highly stochastic and non-deterministic which makes the task of modeling human motion very challenging. (Martinez et al., 2017). Specifically, many future poses are of high probability for the same set of observed sequence poses, thus making the task of long-term prediction very complex and non-trivial (Pavllo et al., 2018). Commonly, the literature refers to the task of long-term motion prediction as *motion generation* which is of special interest to the computer graphics community specifically for the animation industry. On the other hand, the task of short-term motion prediction is commonly referred to as *motion prediction* which mostly concerns the community of computer vision. (Pavllo et al., 2018). The former is harder to validate quantitatively, therefore a qualitative metric, specifically human judgment is crucial while the latter can be validated quantitatively typically by measuring the mean-squared error in the angle space (Martinez et al., 2017).

1.2 Recurrent Neural Networks and Motion Prediction

Deep learning methods have proved to be successful for many tasks including pattern recognition and human motion prediction (Pavllo et al., 2019). Particularly, Recurrent neural networks (RNNs) have shown good performance in predicting future 3D human poses not only in the short-term motion prediction (Fragkiadaki et al., 2015) but also in the long-term motion generation (Martinez et al., 2017).

Unlike traditional methods which require expert knowledge about human motion upfront, RNNs like other deep learning methods can be trained to automatically learn representations that generalize to novel tasks depending on the used network structure and the task at hand (Butepage et al., 2017). RNNs excel at processing sequential data because of their ability to capture temporal dependencies between elements of a sequence.

Unlike traditional neural networks known as feedforward neural networks (FNNs), RNNs have internal loops to persist information allowing them to remember the context of previously seen inputs.

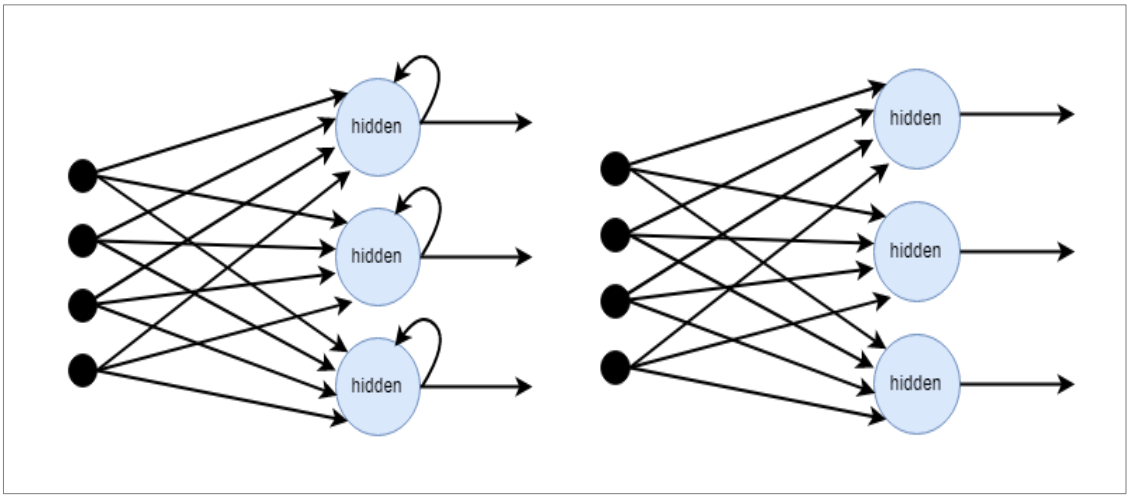

Figure Error! No text of specified style in document..**2: RNN vs FNN.**
(left) Recurrent neural network (RNN). (Right) Feed forward neural network (FNN). Figure adopted from (Mulder et al., 2015).

FNNs process inputs at each iteration independently meaning that there is no context to be preserved. i.e. FNNs lack any form of memory. In the context of language modeling, machine translation as an example, FNNs can take a fixed number of previous words to predict the incoming word, a drawback that results from how FNNs are built. As a result, all words seen in previous iterations are forgotten even though those necessary to detect the next word. However, RNNs theoretically can preserve arbitrary context lengths. (Mulder et al., 2015). Figure 1.2 illustrates how RNNs differ from FNNs. RNNs will be explored in depth in chapter 3.

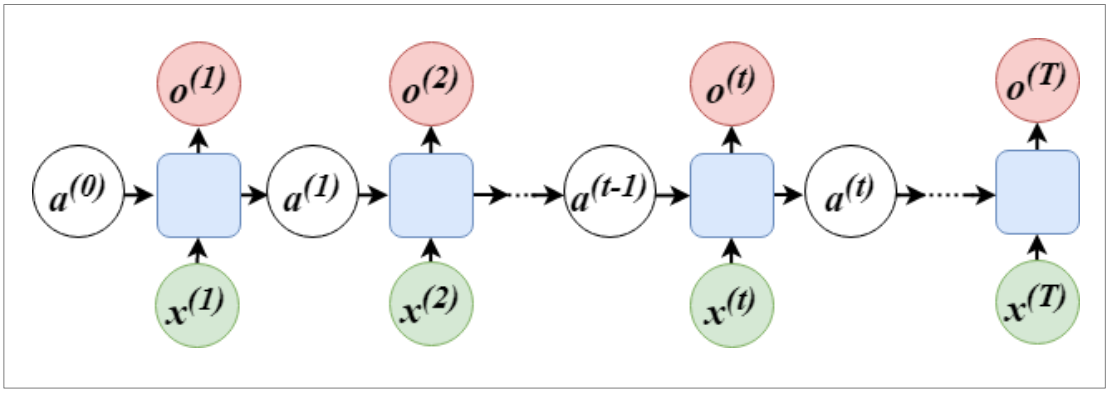

Figure Error! No text of specified style in document..**3: Unidirectional RNN.**

Some applications may require information not only from the past of the input sequence but also from the future. One can consider the case of speech recognition, if there happens to be a word with two different interpretations that both look plausible, it might be crucial to take into account future words alongside with past words to determine the current word (Goodfellow et al., 2016). Bidirectional RNN (BRNN) was invented to fulfill that need (Schuster & Paliwal, 1997).

In a nutshell, BRNN combines two RNNs, one of them processes the sequence in the forward direction starting from the beginning of the sequence (forward through time) while the other one processes the sequence in the backward direction starting from the end of the sequence (backward through time) (Goodfellow et al., 2016). Figures 1.3 and 1.4 show how a unidirectional RNN differs from a bidirectional RNN. BRNNs will be discussed further in chapter 3, section 3.9.

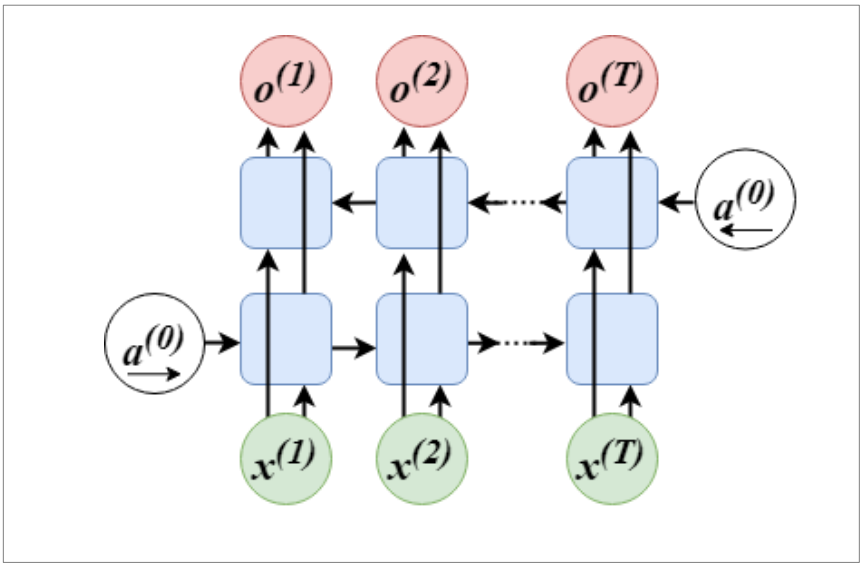

Figure Error! No text of specified style in document..**4: Bidirectional RNN (BRNN). Figure adapted from (Amidi & Amidi).**

1.3 **Motion Data**

The task of modeling human motion relies often on motion data acquired through motion capture technology. Mainly, Motion capture (MOCAP) is a cost-effective technique used to acquire highly realistic motion data by recording the performance of live actors to be

converted later to mathematical representation and consequently applied to a mathematical model (Dean, 2016). In general, MOCAP uses an articulated model or *skeleton* that expresses the human joint chain which imposes certain parent-child relationships between joints (Wang et al., 2014). Among several MOCAP datasets that are publicly available (Sigal et al., 2009; Ionescu et al., 2014; Carnegie Mellon University - motion capture library), Human3.6M is currently the largest dataset with 3.6 Million accurate 3D human poses obtained by recording 15 different activities of 11 professional actors using accurate marker-based motion capture system namely Vicon system (Ionescu et al., 2014). Two common pose parametrizations considered in the literature are provided by the Human3.6M including *relative 3D joint positions representation* and *Kinematic representation* with a full skeleton of 32 joints for both representations (Ionescu et al., 2014; Ionescu et al., 2011). As in previous works of Martinez et al., (2017) and Fragkiadaki et al., (2015), Human3.6M dataset is used in this work to assess quantitatively the use of bidirectional recurrent Neural Networks (BRNNs) in human motion prediction task.

1.4 Overview of Motion Capture Data

Conventional methods of character animation are very time consuming and require skillful animators to pose a character with the aid of specialized animation software. As a result, a need for better alternatives arises to fulfill the increasing demands of the animation industry. MOCAP technology provides a solution that creates better looking 3D animation in a shorter amount of time.

A comprehensive definition is given by Dyer, Martin, & Zulauf, (1995) which states that MOCAP:

> *involves measuring an object's position and orientation in physical space, then recording that information in a computer-usable form. Objects of interest include human and non-human bodies, facial expressions, camera or light positions, and other elements in a scene.*

The process of capturing motion is accomplished by using either physical information or image information provided by sensors to reconstruct the joints of the skeleton. Depending on the used techniques, MOCAP could be classified to sensor-based and image-based motion capture (Mulder et al., 2015). Sensor-based MOCAP involves the use of physical sensors including but not limited to inertial, optical and pressure sensors whereas image-based MOCAP involves the use of single or multiple cameras to capture human motion by acquiring information from colored or depth images (Mulder et al., 2015). An extensive presentation of different motion capture techniques can be found in (Mulder et al., 2015). Another classification divides MOCAP into marker-based and marker-less depending on the tracking technology (72.b Documentation with motion capture, 2012). Generally, marker-based MOCAP offers a way to acquire animations directly from live actors by attaching markers to the body of the subject who wears a motion-capturing suite as shown in Figure 1.5, and then record the motion by tracking key points in the space over time and finally, converting them to 3D digital form (Meredith & Maddock, 2001). Figure 1.5 shows a set of markers being attached to the actor's body. The less common alternative is the marker-less MOCAP which doesn't involve the use of any artificial enhancements of the object or the environment other than the sensors (72.b Documentation with motion capture, 2012).

Figure Error! No text of specified style in document..**5: Marker-based MOCAP.**
An actor wearing a motion-capturing suite with a set of markers attached to his body (Carnegie Mellon University - motion capture library - info).

1.2.3 **MOCAP File Formats**

Before mentioning the most common MOCAP file formats, it would be useful to define some terminologies that are necessary to understand these formats.

Skeleton: The entire character that motion data is applied to make animation. The skeleton consists of a set of bones. Figure 1.6 shows a hierarchical skeleton alongside the hierarchy of the bones (Meredith & Maddock, 2001).

Bone or joint: The smallest entity in the motion that is subject to individual translational or/and rotational changes throughout the animation. Bones are connected by joints which are related to each other by a parent-child relationship. The movement of the joints down the hierarchy i.e. children joints, is affected by the movement of joints higher in the hierarchy i.e. parent joints (Meredith & Maddock, 2001). The root joint as illustrated in Figure 1.6 is the hip joint. The hierarchical structure of the of the bones is shown to the left of Figure 1.6. Different datasets have different number of joints. For example, the number of joints composing the skeleton of Human3.6M is 32 while it's 57 in CMU dataset.

Degree of freedom / channel: Translation and rotation changes can be applied to joints over time to generate animation. These changes represent degree of freedoms (DOFs) of joints. Usually, a joint may have between 1 to 6 DOFs (Khan, et al., 2017).

Frame: Any animation is composed of a set of frames that when played consecutively generate motion. A single frame of motion contains channel DOF data for every bone in the skeleton (Meredith & Maddock, 2001).

MOCAP data comes into different formats. One of the most common formats is Biovision Hierarchy (BVH) format with .bvh extension. It was developed originally at Biovision and gained its popularity since then due to its simplicity (72.b Documentation with motion capture, 2012).
Any BVH file is comprised of 2 sections, the hierarchy section which contains information about the hierarchy of the skeleton and the initial pose as shown in Figure 1.7 and the

motion section which contains information about the channels for every joint as shown in Figure 1.8.

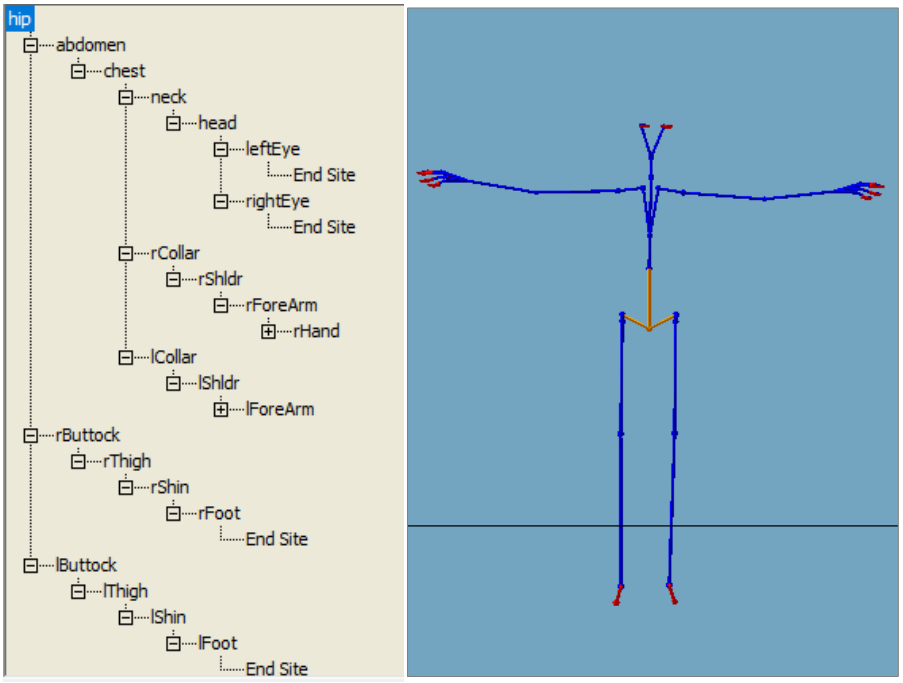

Figure Error! No text of specified style in document.**.6: An articulated body model or skeleton illustrates the hierarchy of bones.**
The root i.e. hip bone is highlighted in orange.

Specifically, the hierarchy section starting with HIERARCHY keyword is used to determine the structure of the skeleton based on the joint's hierarchy. ROOT keyword that comes afterwards refers to a common root joint which indicates the start of the skeletal structure. Following the root is a chain of joints connected to each other with a parent-child relationship and encapsulated within a pair of curly braces where each child joint is proceeded with JOINT keyword.

However, the end of the chain is indicated with (End site) keyword which refers to an end-effector that is, the last joint in a chain with no children where offset values of this joint indicates the bones' length and orientation.

Each joint has a position indicated by the three numbers following OFFSET keyword. These numbers represent x y z relative positions or translations of a joint with respect to its parent. In the case of the root joint, these numbers represent a global position.

Furthermore, offset values determine implicitly the length and orientation of the parent bone. CHANNELS keyword specifies the DOFs of a joint. The order of channels for every joint matches the order of data in the motion section of the file. Particularly, the first 6 values in the motion section correspond to the channels defined for the root joint in the hierarchy section in the same order they appear with and the next 3 values correspond to the channels of the child joint that comes next in the hierarchy in the same order they appear with and so forth.

```
HIERARCHY
ROOT hip
{
  OFFSET 0 0 0
  CHANNELS 6 Xposition Yposition Zposition Zrotation Yrotation Xrotation
  JOINT abdomen
  {
    OFFSET 0 20.6881 -0.73152
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT chest
    {
      OFFSET 0 11.7043 -0.48768
      CHANNELS 3 Zrotation Xrotation Yrotation
      JOINT neck
      {
        OFFSET 0 22.1894 -2.19456
        CHANNELS 3 Zrotation Xrotation Yrotation
        JOINT head
        {
          OFFSET -0.24384 7.07133 1.2192
          CHANNELS 3 Zrotation Xrotation Yrotation
          JOINT leftEye
          {
            OFFSET 4.14528 8.04674 8.04672
            CHANNELS 3 Zrotation Xrotation Yrotation
            End Site
            {
              OFFSET 1 0 0
            }
          }
          JOINT rightEye
          {
            OFFSET -3.6576 8.04674 8.04672
            CHANNELS 3 Zrotation Xrotation Yrotation
            End Site
            {
              OFFSET 1 0 0
            }
          }
        }
      }
    }
```

Figure Error! No text of specified style in document.**.7: The hierarchy section of a BVH file with (.bvh) extension.**

```
MOTION
Frames: 3
Frame Time        0.03333
-71.6924          85.1887          0.966493          0.0               0.0               0.0
0.559351         -0.00286671       0.0159889         0.154366         -0.00442833        0.0267461
3.14654e-15       7.95139e-16      -9.31803e-17       9.44848e-15       3.18055e-15      -2.64011e-16
-0.0              0.0               0.0              -0.0               0.0               0.0
1.63875e-15       7.76503e-18       1.49089e-16       3.37746e-14       6.21202e-18       0.0
5.67431e-13      -1.62208e-13      -2.22639e-14       6.46653e-12       2.0276e-13       -1.90833e-13
1.52547e-11       9.0012e-14       -4.45278e-13       2.63322e-11       1.55847e-13      -7.62538e-13
1.52555e-11       8.78628e-14      -4.51639e-13       7.73091           5.47488           0.417314
1.52531e-11       8.94531e-14      -4.42097e-13       2.90554           0.965183          0.102902
1.52569e-11       8.94531e-14      -4.38121e-13       10.2786           0.731801          0.0259123
1.52454e-11       9.26337e-14      -4.23014e-13       2.22167          -0.733487         -0.20892
1.34313e-15       6.60027e-18      -2.48481e-16       3.14461e-14       6.21202e-18      -1.59028e-15
5.59927e-13       1.75129e-13      -2.8625e-14        6.46289e-12      -1.37186e-13      -1.62208e-13
1.52401e-11       8.48947e-14      -4.23014e-13       2.63338e-11       1.49486e-13      -7.43455e-13
1.52384e-11       8.26944e-14      -4.16653e-13      -7.73091           5.47488          -0.417314
1.52414e-11       8.39865e-14      -4.35736e-13      -2.90554           0.965183         -0.102902
1.52414e-11       8.48811e-14      -4.33351e-13      -10.2786           0.731801         -0.0259123
1.52502e-11       8.74653e-14      -4.48458e-13      -2.22167          -0.733487          0.20892
-2.03217e-16      0.0              -3.88251e-19       0.922987         -4.50513          -0.0770314
-0.00206825       2.60853          -0.00962928       -0.502454          1.89677           0.0959411
2.03217e-16       0.0               3.88251e-19      -0.0712528        -2.35215          0.0224559
0.008072         -1.29119          0.069199          0.494623          3.64342          -0.130519
```

Figure Error! No text of specified style in document.**.8: The motion section of a BVH file with (.bvh) extension.**
Channel information of the first frame of motion is shown

The MOTION section of the file starts with number of frames and frame's duration or frame rate and finally the channel data for each bone as the appear in the hierarchy section of the file. In fact, the channel data is nothing more than the animation data of each bone through time. An older MOCAP format is the Biovision Action File (BVA) with .bva extension which is similar to BVH file in many ways but with key differences, the most important of which is that BVA can't store motion for a hierarchical skeleton. The motion of a child bone doesn't depend on the motion of any other joints (Lander, 1998). Another common format is the Hierarchical Translation Rotation (HTR) format with .htr extension which was developed by Motion Analysis company with the aim of solving some problems of BVH files (Du et al., 2016). Acclaim motion file is yet another popular format designed by Acclaim gaming company. This format consists of two files. The first file is the Acclaim Skeleton File ASF with .asf extension which contains information about the hierarchy and the initial pose of the skeleton. The second file is the Acclaim Motion Capture (AMC) file with .amc extension which is used to save motion data of the skeleton. This separation between the hierarchy information and the motion data is useful since only one ASF file can be used in a motion capture session with multiple AMC files (Lander, 1998). Coordinate 3D (C3D) is a binary format defined by the National Institute of Health mainly

to be used in the biomechanical research. Unlike previously described formats which merely contain information about 3D positions and orientations, C3D supports wide diversity of data that is useful for the biomechanical research (72.b Documentation with motion capture, 2012).