Virtual Radar Spectrograms for Skeleton Based Action Recognition

Heewon Hah
Dept. of Computer Science
University of North Carolina at Charlotte
Charlotte, North Carolina - 28262
hhah@uncc.edu

Kalvik Jakkala
Dept. of Computer Science
University of North Carolina at Charlotte
Charlotte, North Carolina - 28262
kjakkala@uncc.edu

***Abstract*—The fundamental problem of skeleton-based action recognition lies dormant and stagnated with approaches limited to incremental approaches suitable only in certain use cases. We propose a theoretically grounded method to convert skeleton graphs to spectrograms by simulating a virtual spectrogram. Our approach simultaneously solves the crucial limitations of the current SOTA. The spectrograms generated with our technique can be used as a virtualization method and accommodate a consistent data representation in the presence of varying frame rates and a varying number of people while maintaining highly correlated feature-rich representations suitable for Convolutional Neural Networks. We establish the merits of our approach by benchmarking it against current techniques on the NTU dataset.**

1. Introduction

Action recognition, the task of recognizing a person's actions, is a well-studied computer vision problem. Action recognition is performed by humans every day and has many potential use cases in numerous fields such as robotics, health care, and surveillance. As promising as it is, action recognition also has severe privacy concerns as most of the methods rely on video data. The privacy concerns have been the primary motivating factor of skeleton-based action recognition. Skeleton based action recognition, unlike its image-based counterpart, does not need the video data once a skeleton is generated from it.

Moreover, skeleton-based action recognition has facilitated the development of action recognition methods agnostic to the medium of data collection. Apart from privacy concerns, it also circumvents numerous biases introduced in data such as race and gender. Although there is a rich literature concerning biases in data and ways to ameliorate them, skeleton-based action recognition entirely mitigates the issue by removing a majority of the biases in such data. Moreover, unlike video, skeleton data is exceptionally sparse yet maintains the features relevant for action recognition.

Now, as promising as it sounds, skeleton-based action recognition also comes with its limitations, that being the sparsity of the data. Convolutional neural networks (CNNs), the mainstay of deep learning, owe their success to the kernels capable of exploiting dense feature correlations in images. But, skeleton data is nothing but feature-rich. As we mentioned before, although their sparsity is a boon for privacy and storage requirements, they fall short when it comes to the actual classification task. Consequently, two predominant approaches have come about to address the limitation mentioned above: graph neural networks (GNNs) and feature representations suited for CNNs. GNNs attempt to classify skeleton data by interpreting the skeleton sequence as a graph. In contrast, feature representation methods craft dense feature representation from the sparse skeleton data and use CNNs for classification. Both approaches, however, are riddled with issues.

GNNs have risen to prominence in recent years, but they are nowhere on par with CNNs in classification performance. And there is no clear consensus about a standard approach to dealing with the temporal information in skeleton data; some treat it with LSTMs while others consider it a part of a static graph. Neither approach has gained a foothold as the de facto standard for treating such data, and both have shown inconsistent classification performance. On the other side of the spectrum, the methods designed to fabricate dense feature-rich representations from sparse skeleton data fail to utilize the entire feature space. Although such representations are denser compared to skeleton data, most approaches only use a fraction of the feature space and consider arbitrary transformations of skeleton data to represent it as an image without a proper theoretical explanation. Moreover, it introduces the added complexity of visually interpreting the data. Once the skeleton data is converted to dense feature representations, it is difficult for humans to analyze it further. Much like GNNs, dense feature representations fail to perform consistently across benchmarks. Finally, both methods are severely limited when dealing with multiple skeletons in a scene.

Therefore, we propose a theoretically grounded approach capable of generating dense feature representations for skeleton data, which maintains a consistent temporal and spatial data representation while being visually interpretable. We do this by simulating a virtual radar and computing spectrograms generated by the skeletons. Such spectrograms maintain consistent representations for skeleton data at varying frame rates and the number of skeletons in a given

scene. We show that such an approach is well suited for skeleton-based action recognition while addressing all the contemporary art's aforementioned limitations.

We begin by presenting the related works in section 2. In section 3, we detail some of the background information required to understand our approach, after which we set-forth the proposed method in the subsequent section. We present our experimental results in section 5 and conclude in section 6.

2. Related Works

3. Preliminaries

Before we can detail the proposed method, we need to understand how radars work and how spectrograms generated from radar data can be interpreted.

Radars, unlike cameras, are active sensors, i.e., they actively interact with the environment to perceive it. Radars achieve this by actively transmitting electromagnetic waves in a periodic fashion called chirps. The chirps bounce off the environment until a few of the signals make their way back to the receiver, collocated with the radar's transmitter. Based on the time difference of a chirp's trip from the transmitter to the receiver, one can estimate an object's location in space. Although a point cloud can be extracted from radar data, radars intrinsically generate signals represented as complex values that are then processed to extract 3D coordinates or spectrograms. We suggest [] for an in-depth exposition of radars and the pertaining theory.

Spectrograms, are the conventional method used to visualize radar signals. When represented as a spectrogram, radar signals map an object's velocity to the vertical axis and time to the horizontal axis. The pixel intensity represents an object's energy, which can also be interpreted as the amount of signal reflected by an object. The pixel intensity is crucial here, as different materials absorb and reflect varying amounts of radar signals. When a signal bounces around in the environment for long periods of time before returning to the radar, its energy decays proportionally. Such an approach is well suited for action recognition, as spectrograms maintain a dedicated temporal axis, allow for a single image representation when multiple people are in a scene, and accommodate varying frame rates for each skeleton in a data sample.

4. Method

Here, we discuss our work's crux, the virtual radar layer, which is used to convert skeleton data to spectrograms.

To generate a spectrogram from radar data, one need not use a physical radar. There are well-established mathematical models [1] of radar signal propagation, which can calculate radar signals generated by hypothetical objects in space. Such models are unfortunately limited to rudimentary geometric shapes such as spheres and ellipsoids, but skeleton data can be represented by those spheres and ellipsoids. The

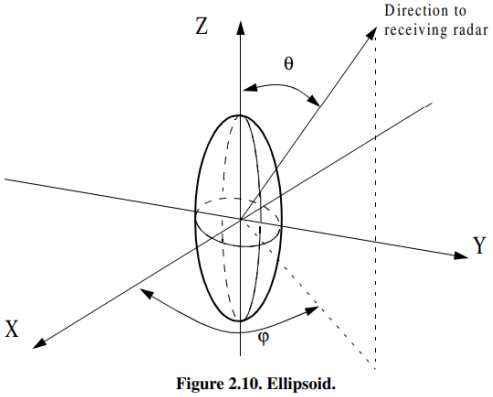

Figure 2.10. Ellipsoid.

Figure 1. RCS model

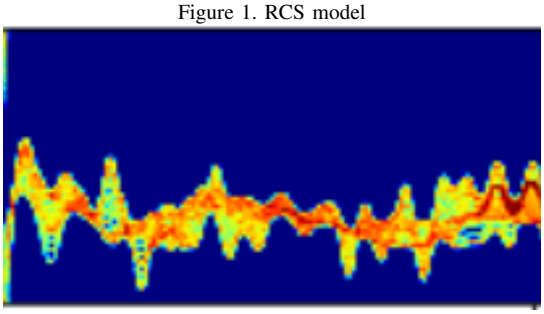

Figure 2. Spectrogram

skeleton can therefore then be used to generate a virtual radar signal and the corresponding spectrogram. Concretely, an ellipsoid is defined by

$$\left(\frac{x-x_0}{a}\right)^2+\left(\frac{y-y_0}{b}\right)^2+\left(\frac{z-z_0}{c}\right)^2=1, \quad (1)$$

where (x_0, y_0, z_0) represents the center of the ellipsoid and a, b, c are the semi-principal axes lengths. To create realistic skeleton data, each body segment length is estimated using average body ratios with respect to the body height. To get the walking movement of the skeleton data, the forward walking movement of each body segment is calculated as the changing positions of the body joints using joint translation, rotation, and flexing equations [2].

Now, the mathematical model of the radar signals reflected, also known as radar backscatter (RCS), by such ellipsoids is given by

$$RCS=\frac{\pi a^2b^2c^2}{((a\sin\theta\cos\phi)^2+(b\sin\theta\sin\phi)^2+(c\cos\theta)^2)^2}. \quad (2)$$

Here θ is the angle between the z-axis of the ellipsoid and the radar's receiver direction. Similarly, ϕ is the angle between the x-axis of the ellipsoid and the radar's receiver direction [3]. Given the radar reflections, the radar data's complex value representation is computed as follows

$$Phase = \sqrt{RCS} \times e^{\frac{-j4\pi d}{\lambda}}. \quad (3)$$

Here, d is the 2-norm distance from the center of the ellipsoid to the radar, and λ is the radar wavelength. Finally, a Short-Time Fourier Transform (STFT) is applied to the radar signals to generate a spectrogram. The generated spectrogram would be consistent with one generated from real-world radar signals bounced off an ellipsoid.

The skeleton data is represented as a spectrogram by first converting the edges in a skeleton to ellipsoids. Radar reflections of each ellipsoid are then computed and summed to generate a coherent radar signal that could have been observed in the real-world. The resultant radar signals are converted to a spectrogram with an STFT.

Although the spectrograms generated with the method above are well suited for action recognition, it requires the tuning of two hyperparameters - the radar wavelength and radar location relative to the skeletons. The two parameters can be tuned using a validation set, but it is not the only available solution. Every operation used to generate a spectrogram from skeleton data is entirely differentiable. We found that by allowing the gradients from the downstream classification task to propagate back into the RCS model, the two hyperparameters can be learned from the training dataset. Apart from the two hyperparameters, the Fourier basis of the STFT operation can also be learned from the data. The generated spectrograms can be treated as images suitable for classification or regression tasks with any standard CNN. Indeed, we utilized a residual neural network prepended by our virtual radar layer. The network was trained using an Adam optimizer with cross-entropy loss for action classification.

5. Experiments

We ran several experiments to establish the efficacy of our approach. We detail them here in this section along with the implementation details.

Before creating the spectrograms, we visualized the skeleton data itself by animating the walking skeleton data in MATLAB. This was mainly for debugging purposes.

Now, for the virtual radar to provide clear and consistent spectrograms, we need skeleton data sampled at a very high frame rate. We addressed this issue by imputing extra frames computed with linear interpolation on the skeleton data. We implemented and tested our approach first in MATLAB, then in PyTorch, wherein we used nnAudio's STFT layer, which supports learnable Fourier basis.

The virtual radar layer was validated by collecting real-world skeleton data from a Microsoft Kinect Azure camera. Data from a TI mmWave 1843 radar was simultaneously collected. We found the virtual radar approach consistent with the spectrogram generated from the physical radar in our experiments.

6. Conclusion

Although the task of skeleton-based action recognition has been well studied for several years, most of the solutions fail to address at least one crucial facet of the problem. We presented a theoretically sound approach to generating dense feature representations. By simulating a virtual radar, we represent any complex scene with skeletons as a single spectrogram with rich feature correlations that can be exploited by CNNs, while solving contemporary art's fundamental limitations such as visual interpretation and handling multiple skeletons in a scene. We also established our approach's merits by comparing the method to current SOTA on the NTU benchmark. Although we focused our efforts on one particular problem, we believe our approach can be utilized in several other image processing problems. We leave such endeavors for our future work.

References

[1] B. R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*. Boca Raton, FL: Chapman & Hall/CRC, 2000.

[2] R. Boulic, N. Magnenat-Thalmann, and D. Thalmann, "A global human walking model with real-time kinematic personification," *The Visual Computer*, vol. 6, pp. 344–358, Nov. 1990.

[3] R. R. Krishna, R. M. Krishna, R. G. Krishna, and D. Sekhar, "Radar cross section prediction for different objects using mat lab and radar cross section (rcs) reduction," *International Journal of Advanced Research in Computer Science and Electronics Engineering*, vol. 1, no. 5, pp. 67–75, Jul. 2012.