Human Pose Estimation and Shape Modeling in 3D: New Cameras, Datasets and Approaches

by

Shihao Zou

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Engineering

Department of Electrical and Computer Engineering

University of Alberta

Abstract

Human pose estimation and shape modeling serve as critical elements in a wide range of computer vision applications. While most existing research employs RGB cameras for their accessibility and cost-effectiveness, emerging camera technologies and imaging modalities are relatively underexplored. These novel technologies often introduce unique features that can provide new avenues for advancement in the fields of human pose estimation and shape modeling. Therefore, this thesis aims to investigate human pose estimation and shape modeling from and particularly beyond RGB cameras by exploring the potential opportunities presented by emerging camera technologies. Our research is organized into three key areas: the exploration of new cameras, the development of novel approaches, and the creation of large-scale multi-modality datasets for human pose estimation and shape modeling.

1) Our research in 3D skeletal pose estimation, tracking, and motion forecasting for multi-person scenarios using RGB cameras addresses complexities like intra-frame occlusions. We propose a unified spatiotemporal transformer with spatiotemporal deformable attention to simultaneously execute these tasks in one computational pass. 2) We further explore event cameras, innovative sensors that balance high temporal resolution with low energy consumption, for energy-efficient parametric shape estimation and tracking. Our approach includes a two-stage deep learning method that primarily uses event data, initially requiring only the first gray-scale frame, and later, an end-to-end approach using Spiking Neural Networks (SNNs) for efficient pose track-

ing from events alone. 3) Utilizing polarization cameras, which capture robust geometric surface cues, we propose a framework for reconstructing detailed, clothed human shapes, beyond skeletal poses or basic parametric shapes. 4) Finally, we turn our focus to the complex task of animating clothed humans with natural clothing deformations, leveraging point-cloud sequences captured by depth sensors that provide valuable geometric insights into the structure of the clothing. We introduce a diffusion-based method for clothed human modeling that integrates dynamics, progressive, and diversified modeling, addressing gaps in current data-driven approaches.

To overcome the limitations of existing datasets primarily based on RGB cameras, we developed a cost-effective motion capture system that synchronizes multi-modality cameras and a pipeline for annotating 3D parametric pose and shape. This led to the creation of several large-scale datasets for human pose estimation and shape modeling: 1) PHSPD with 527K frames featuring polarization and multi-view RGB-Depth images, 2) MMHPSD with 240K frames containing event streams and RGB-Depth images, and 3) SynEventHPD, a synthesized event-based dataset. Together, PHSPD, MMHPSD, and SynEventHPD form the most extensive and varied 3D human motion capture datasets available, with their multi-modality property holding significant potential for driving existing and new research directions in the computer vision community.

In summary, this thesis demonstrates that emerging camera technologies such as polarization cameras, event cameras, and point-clouds provide new perspectives and effective solutions for related tasks in the fields of human pose estimation and shape modeling. Extensive experiments across various projects further validates the effectiveness of the novel approaches we propose for these tasks.

Preface

This thesis is an original work by Shihao Zou.

Chapter 3 of this thesis has been published under the title: [268] **S. Zou**, Y. Xu, C. Li, L. Ma, L. Cheng, and M. Vo, "Snipper: A spatiotemporal transformer for simultaneous multi-person 3d pose estimation tracking and forecasting on a video snippet," IEEE Transactions on Circuits and Systems for Video Technology (IEEE TCSVT), 2023. I implemented the technical design with the assistance of Y. Xu, C. Li, L. Ma, and M. Vo. The experiment results and analysis are my original work.

Chapter 4 of this thesis has been published under the title: [266] **S. Zou**, C. Guo, X. Zuo, S. Wang, P. Wang, X. Hu, S. Chen, M. Gong, and L. Cheng, "Eventhpe: Event-based 3d human pose and shape estimation," in IEEE/CVF International Conference on Computer Vision (ICCV), 2021. I led the dataset capture with the assistance of C. Guo, X. Zuo, S. Wang, X. Hu, S. Chen, M. Gong, and L. Cheng. I also primarily implemented the technical design and experiments, with support from C. Guo and P. Wang.

Chapter 5 of this thesis has been submitted for review and currently preprinted under the title: [267] **S. Zou**, Y. Mu, X. Zuo, S. Wang, and L. Cheng, "Event-based human pose tracking by spiking spatiotemporal transformer," arXiv preprint arXiv:2303.09681, 2023. I led the dataset synthesis with the assistance of Y. Mu, X. Zuo and S. Wang. I also primarily implemented the technical design and experiments, with support from Y. Mu and L. Cheng.

Chapter 6 of this thesis has been published under the title: [269] **S. Zou**, X. Zuo, Y. Qian, S. Wang, C. Xu, M. Gong and L. Cheng, "3d human shape reconstruction from a polarization image," in European Conference on Com-

puter Vision (ECCV), 2020. The following work has been published under the title: [270] **S. Zou**, X. Zuo, S. Wang, Y. Qian, C. Guo, and L. Cheng, "Human pose and shape estimation from single polarization images," IEEE Transactions on Multimedia (IEEE TMM), 2022. I led the dataset capture with the assistance of X. Zuo, S. Wang, C. Xu and C. Guo. I also primarily implemented the technical design and experiments, with support from Y. Qian, M. Gong and L. Cheng.

Chapter 7 of this thesis has been submitted for review and **S. Zou** is the first author of this work. I implemented the technical design with the assistance of co-authors. The experiment results and analysis are my original work.

To the silent moments in the early hours of the morning,
where inspiration struck and perseverance was tested.

We can only see a short distance ahead, but we can see plenty there that needs to be done.

– Alan Mathison Turing

Acknowledgements

I would like to express my deepest gratitude to my advisor, Prof. Li Cheng, for his unwavering support, guidance, and mentorship throughout the course of my PhD journey. Your expertise and enthusiasm for the research in computer vision have been a constant source of inspiration, and your commitment to my growth as a researcher and scholar have been invaluable.

I am grateful to my thesis committee members, Prof. Herb Yang, Dr. Xingyu Li, Dr. Jun Jin and Dr. Leonid Sigal, for their insightful feedback, constructive criticism, and encouragement during the development of this thesis. Your expertise and dedication to the academic community have shaped my work and contributed to the success of this research.

I would like to extend my appreciation to all the members in the Vision and Learning Lab for their camaraderie, collaboration, and stimulating discussions. Special thanks to Wei Ji, Chuan Guo, Jingjing Li, Ji Yang and Yuxuan Mu for their invaluable assistance in the lab and for sharing their knowledge and expertise. I would like to extend my warmest thanks to my mentors, Xinxin Zuo, Sen Wang, Yiming Qian, Minh Vo, Yuanlu Xu, for their invaluable guidance, support, and encouragement throughout my academic journey. Their expertise, wisdom, and mentorship have been instrumental in my growth as a researcher and a professional.

Lastly, I am forever grateful to my family for their unconditional love, support, and sacrifices. To my parents, thank you for instilling in me the value of hard work and perseverance, and for always believing in me. To my girlfriend, Clare Yue, your love, patience, and encouragement have been my rock, and I could not have completed this journey without you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Human pose estimation and shape modeling serve as the backbone for a wide array of applications in computer vision, such as action recognition, bio-mechanics and medical diagnostics, human-computer interaction, autonomous driving, video surveillance, digital human, Virtual Reality (VR) and Augmented Reality (AR) technology [193], [208].

Within specific domains, these technologies assume specialized functions. For example, in the domains of action recognition and bio-mechanics, these techniques assist in the classification and analysis of human activities. Specifically, they offer valuable tools for scrutinizing gait, posture, and joint dynamics, thereby enhancing applications in sports analytics and medical diagnostics. In the realm of video surveillance, they excel in tracking individuals within crowded spaces and identifying unusual activities based on atypical body movements or postures. In the context of human-computer interaction, these technologies have revolutionized user engagement by enabling more intuitive, gesture-based controls, thus eliminating the need for conventional input devices such as keyboards or mice. In the field of autonomous driving, they improve safety by accurately identifying pedestrians, discerning their intentions, and predicting their subsequent movements. Within the digital human space, these technologies facilitate the creation of photorealistic avatars and digital doubles, which are employed in cinematic productions, video games, and virtual social interactions. Finally, in VR and AR experiences, pose esti-

mation and shape modeling contribute to immersive user engagement by creating photorealistic avatars and seamlessly translating real-world movements into virtual environments. This has far-reaching ramifications across a diverse range of fields including education, training simulations, and entertainment.

Over the past decade, significant advancements have been made in the field of human skeleton pose estimation [24], [55], [70], [90], [142], [144], [146], [150], [209], [212], [249], [252]. As depicted in Fig. 1.1 (a), earlier studies commonly employ a skeletal representation, comprised of predefined joints such as shoulders, pelvis, knees, ankles, neck, nose, wrists, and elbows. This skeleton-based parameterization offers a straightforward and effective way to represent human poses in 2D or 3D contexts.

With the advent of parametric models of human pose and shape, such as SMPL [116] and SMPL-X [153], the landscape of pose estimation techniques has expanded considerably [10], [19], [48], [91], [92], [97], [101], [117], [156], [253], [271]. As depicted in Fig. 1.1 (b), these parametric models offer a compact, low-dimensional statistical representation of human shapes, employing only 82 parameters in the case of the SMPL model to define the human shape of a triangular mesh comprising 6890 vertices. Such models are learned on extensive datasets of minimally-clothed human bodies.

In addition to these advancements, the past decade has seen the emergence of deep learning techniques that have revolutionized human pose and shape estimation. Researchers have developed various end-to-end deep learning methods for human pose estimation or parametric shape estimation [24], [55], [70], [91], [92], [97], [142], [144], [150], [156], [209], [212], [249], [252], effectively demonstrating the capabilities of deep learning in this research domain.

Unlike earlier skeletal or parametric representations, recent focus has shifted toward deep learning-based human shape reconstruction [74], [78], [121], [157], [172], [173], [244], [257], [264]. This transition is largely driven by the powerful learning capabilities of deep learning algorithms. As illustrated in Fig. 1.1 (c), where clothed human bodies are inferred from 2D images using end-to-end deep learning methods and represented in diverse forms, such as volumes, point-clouds, or meshes. This representation is usually more attractive as it

(a) Articulate Skeleton Representation of Human Pose

(b) Parametric Representation of Human Pose and Shape

(c) Volume / Point-cloud / Mesh Representation of Clothed Human Shape

Figure 1.1: Three commonly used ways to represent human pose and shape. (a) Articulate skeleton representation of human pose [24]. (b) Parametric representation of human pose and shape [91]. (c) Volumetric, point-cloud or mesh representation of clothed human shape [74], [173].

captures the entire appearance of an individual, including intricate details of clothes.

1.2 Motivations

Most current research in human pose estimation and shape modeling primarily utilizes RGB cameras due to their widespread availability, affordability, and ease of integration into a variety of computer vision applications. However, as new types of cameras and imaging modalities emerge, *there has been limited exploration into the potential opportunities they present.* The unique features of these emerging technologies often provide new perspectives and solutions in our field of study. To this end, we outline some of these camera technologies below.

- RGB cameras are widely available and relatively affordable, making them cost-effective and easy to integrate into plenty of computer vision applications like object detection [65], semantic segmentation [28], and human pose estimation and shape modeling [24], [91], [173]. However, their high bandwidth requirements for transmitting high-frame-rate images limit their effectiveness in low-energy applications. Additionally, they lack direct capabilities for capturing depth or geometric information, leading to ambiguities in 3D vision tasks.

- Event cameras are an emerging category of bio-inspired sensors that produce sparse data, easing both processing and storage demands. These sensors can optionally deliver high-frame-rate gray-scale video as well. These sensors excel in low latency, energy efficiency, and resistance to motion blur. However, they generally come at a higher cost and offer limited color and depth information.

- Polarization cameras, built upon the physical law that reflected light is usually polarized, can provide additional geometric clues and reduce glare from surfaces like water or glass. Despite these advantages, they

are complex, costly, and computationally demanding in terms of data processing.

- Point-clouds, generated by Time-of-Flight sensors such as Kinect [131] or Lidar [177] cameras, provide rich spatial information and are versatile and flexible, making them valuable for a range of applications from 3D modeling to object recognition. However, they come with challenges such as high data volume, sensitivity to sensor noise, and the absence of semantic and topology information.

While imaging techniques have significantly advanced in recent years, it's crucial to recognize that each imaging technology carries its own set of pros and cons, suited to particular applications and constraints. RGB cameras, which are widely used and cost-effective, are adept across various applications. Nonetheless, for industrial needs where low power consumption and real-time performance are paramount, such as in video surveillance or object detection in autonomous driving, event cameras could be a promising alternative. In contrast, while RGB images or event streams fall short in providing accurate depth or 3D information for intricate tasks like human shape modeling, alternatives like polarization images or point clouds might be suitable, albeit with higher costs and lower frame rates. Therefore, each imaging modality brings its own set of trade-offs, necessitating careful selection based on the application's unique demands. Expanding on our exploration of these cutting-edge camera technologies, we propose that integrating diverse sensor types could significantly improve pose estimation accuracy under varied conditions.

Consequently, *this thesis will delve into the realm of human pose estimation and shape modeling from and particularly beyond RGB cameras to explore the potential opportunities presented by these emerging camera technologies.* Specifically, our research in this thesis is structured around three pivotal components: the exploration of new cameras, the development of novel approaches, and the creation of large-scale multi-modality datasets for human pose estimation and shape modeling.

Following the history of human pose and shape representation outlined in

Fig. 1.1, our research begins with 3D skeletal pose estimation and tracking using RGB cameras, as detailed in Chapter 3. This chapter delves into the practical yet challenging task of multi-person pose estimation, tracking and motion forecasting. We then shift our focus to an energy-efficient approach for human parametric shape estimation and tracking in Chapters 4 and 5. Here, we leverage the unique features of event cameras and advance efficient sparse deep learning techniques, *i.e.*, spiking neural networks (SNNs). To reconstruct more detailed and realistic clothed human shapes as opposed to minimally-clothed parametric shapes, Chapter 6 explores the use of polarization cameras, which provide robust geometric clues of human clothing surface details. Lastly, going beyond this static shape reconstruction, Chapter 7 investigates the challenging task of learning to animate clothed humans with natural clothing deformations based on point-cloud sequences, as they offer valuable spatial insights into the geometry of clothing.

Significantly, our research into the use of emerging camera technologies for human pose estimation and shape modeling reveals a notable gap: the lack of publicly available, specialized datasets. To address this, another major contribution of this thesis is the creation of three multi-modality datasets. We have developed an in-house motion capture system that synchronizes multi-view RGB-Depth cameras with a polarization camera and an event camera. Additionally, we have established a pipeline for annotating 3D parametric human poses and shapes. Further details on these datasets will be provided in the following sections and chapters.

1.3 Overview

1.3.1 Multi-Person Pose Estimation from RGB Cameras

While the field has extensively studied single-person pose estimation [91], [97], [142], [252], multi-person skeletal pose estimation, tracking, and motion forecasting from RGB videos are of greater practical relevance but also pose increased challenges due to intra-frame occlusions. Existing methods [24], [52],

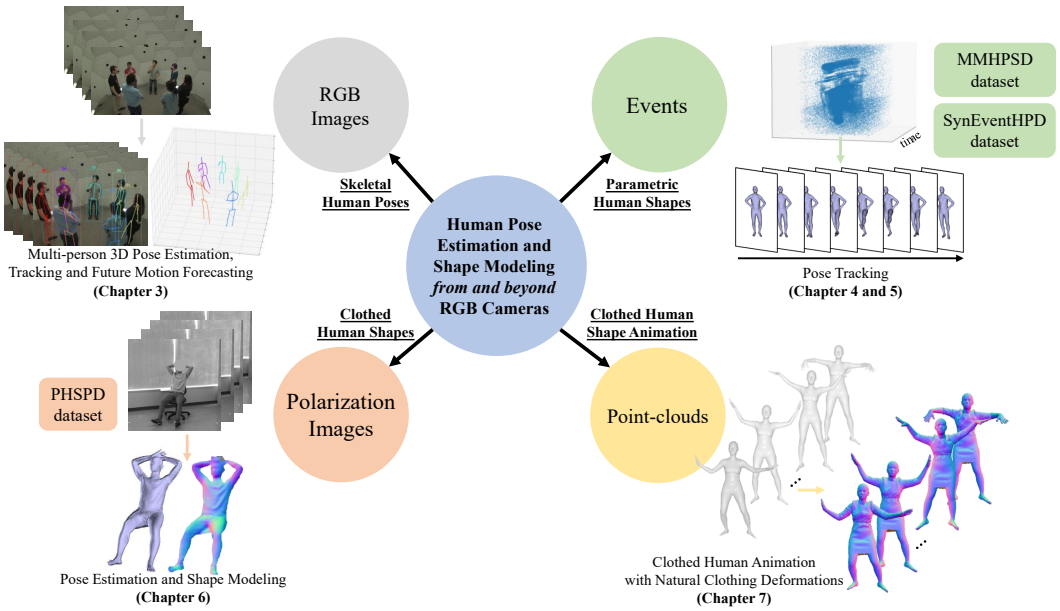


Figure 1.2: Overview of the thesis. We focus on human pose estimation and shape modeling *from and particularly beyond* RGB cameras to explore the potential opportunities new cameras present.

[222] generally concentrate on a single task, or isolate each task and employ multi-stage strategies to solve each task individually. However, pose estimation and tracking are intrinsically correlated: accurate 3D pose estimation facilitates tracking, while robust tracking provides informative temporal clues for pose estimation within the video, particularly when persons are occluded in a single frame. Additionally, tracking accumulates vital historical data that facilitates accurate future motion prediction. Prior studies often overlook these correlations, resulting in sub-optimal decisions at each stage of these inter-connected tasks.

To overcome the aforementioned challenges, Chapter 3 introduces a unified framework designed to concurrently estimate, track, and forecast multi-person 3D poses from RGB video snippets. Utilizing our proposed spatiotemporal deformable attention, this framework effectively encodes spatiotemporal relationships between frames, thereby mitigating the issue of intra-frame occlusions common to RGB cameras. Experimental results reveal that our model outperforms specialized baseline models, demonstrating competitive effectiveness across all three tasks: pose estimation, tracking, and motion forecasting.

This work has been published as [268].

1.3.2 Parametric Shape Tracking from Event Cameras

Different from traditional frame-based cameras, event cameras [59] have been an emerging bio-inspired imaging sensor that bypasses the usual trade-off between high temporal resolution and low energy consumption. In the context of frame-based cameras, high temporal resolution is typically synonymous with a high frame rate. The most important concept of event cameras is an "event", which is defined as a triplet $(\mathbf{x}, t, p)$, indicating a significant brightness change at a specific pixel location $\mathbf{x}$ at time t with its binary polarity as p. Events are triggered only when brightness changes exceed a preset threshold. Rather than capturing images at a fixed frame rate, events are asynchronously registered at per-pixel level in event cameras. The stream of events is also spatially much sparser compared to conventional frame-based cameras, where each image is densely packed with a full stack of per-pixel values. Consequently, event cameras excel at capturing local motions as a series of sparse, asynchronous events. It should be noted that event cameras can optionally provide high-frame-rate gray-scale video as well. With their unique advantages—high temporal resolution, low latency, low power consumption and high dynamic range—event cameras have found applications in various computer vision tasks. These include tracking [63], [135], [245], [246], recognition [3], [57], [95], 3D reconstruction [166], [248], and a wide array of applications in robotics, virtual and augmented reality, and autonomous driving [59].

The potential of event signals in estimating 3D human parametric pose and shape has been rarely explored. Prior method [227] depends not only on events data but also on an auxiliary gray-scale video for initial pose estimation at each time step. This limitation prompts us to explore the feasibility of using events as the primary input source for estimating 3D human poses over time in Chapter 4. In this chapter, we utilize only the first gray-scale frame generated by event cameras to extract the beginning pose and shape. Subsequently, events data is employed to track the following poses and shapes over time. We propose utilizing optical flow inferred from events to reduce

dependency on the entire gray-scale video as additional input. We also introduce a novel coherence loss to ensure alignment between event-based flow (optical flow) and shape-based flow (movements of vertices on human shapes). Our empirical results show that our proposed approach outperforms several state-of-the-art baselines. Additionally, we provide the Multi-Modality Human Pose and Shape Dataset (MMHPSD), featuring 240k frames across multiple imaging modalities, including event cameras. To our knowledge, MMHPSD is the largest event-based 3D human pose and shape dataset and the first to be publicly available. Its multi-modality property enhances its potential in facilitating existing and new research directions. This work has been published as [266].

Furthermore, existing methods either require the presence of additional gray-scale video or frame [227], [266], which may not be feasible in practice, or treat the event stream as frame-based dense images [23], [170] and input them directly into the ANNs models, which ignores the inherent sparsity of event signals. Motivated by the above observations, our subsequent work aims to tackle a relatively new problem of tracking 3D human poses solely based on event streams from an event camera, thus completely eliminating the need for additional dense images as the input. To address this problem, Chapter 5 introduces a novel end-to-end sparse deep learning approach, which is entirely built upon Spiking Neural Networks (SNNs), thus having the promise of being more efficient than the dense deep learning models built upon ANNs. Extensive empirical experiments demonstrate the superior performance of our approach over existing state-of-the-art methods [227], [266] and several ANNs baselines [26], [98], [115]. Additionally, this is achieved by utilizing merely around 20% of the computation (in FLOPs) and 3% of the energy consumption required by the ANNs baselines. Additionally, a large-scale dataset, SynEventHPD, is constructed for the task of event-based 3D human pose tracking. It consists of synthesized events data from multiple motion capture datasets and consequently covers a variety of motions with a total size of 45.72 hours event streams – more than 10 times larger than MMHPSD [266], the largest existing event-based pose tracking dataset. This subsequent work has been published

as [267].

1.3.3 Human Shape Reconstruction from Polarization Cameras

RGB images usually lack geometric cues of an object's surface, which normally results in the ambiguity of 3D pose estimation or human shape reconstruction. This observation inspires us to investigate a new imaging modality: polarization camera, which means we consider the problem of estimating human pose and reconstructing human shape from a single polarization image. Polarization camera is built upon a basic physics principle: a light ray reflected from an object is usually polarized. This polarized signal thus carries sufficient geometric cues of the object's surface, enabling more reliable inference of surface normals [9], [231]. It is worth mentioning the biological fact that light polarization could be directly perceived by some species of bees, ants, and shrimp for purposes such as 3D navigation [42], [218].

Inspired by these physical and biological observations, Chapter 6 presents a dedicated two-stage method, named HumanSfP, for estimating human pose and shape using geometric cues from polarization images. The first stage, Polar2Normal, focuses on generating accurate surface normal maps from a single polarization image, leveraging relevant physical laws as priors. These predicted surface normals are then used in the second stage, Polar2Shape, to reconstruct a clothed human shape based on an initially estimated parametric shape.

In tackling this new problem, we have created a dedicated dataset, the Polarization Human Shape and Pose Dataset (PHSPD). It consists of ∼527K frames along with corresponding parametric pose and shape annotations. Overall, there are 21 different subjects performing 31 unique actions, and ∼9.5 hours of videos are recorded in total. Empirical evaluations on a synthetic dataset, SURREAL dataset [199], as well as our real-world dataset, PHSPD dataset, demonstrate the effectiveness and applicability of our approach. Our work showcases that, for estimating 3D human poses and shapes, 2D polarization cameras could be a viable alternative to conventional RGB cameras.

This work has been published as [269], [270].

1.3.4 Clothed Human Animation from Point-clouds

With the advancements in hardware technology, depth sensors have increasingly become an accessible tool for capturing 3D point-clouds of objects. In contrast to other 3D formats, such as depth maps, meshes, and voxel grids, point-clouds can be acquired more effortlessly and provide an intuitive, efficient, and general representation of 3D objects.

Leveraging the capabilities of point-clouds, Chapter 7 concentrates on clothed human shapes, which offer a more comprehensive representation than skeletal poses or minimally-clothed parametric shapes, capturing intricate details such as the folds and flow of clothing. These details are essential for applications ranging from filmmaking and game development to virtual and augmented reality. In this chapter, our research extends beyond static human shape reconstruction to focus on the complex task of animating clothed humans with natural clothing deformations. We approach this challenge by utilizing point-cloud sequences, which provide valuable geometric insights into the structure of the clothing.

Specifically, clothed human modeling aims to learn clothing deformation dynamics from a set of 3D point-clouds or meshes of clothed human bodies, facilitating the generation of naturalistic clothing details in target motion animations. This task is inherently challenging owing to the variety of clothes and human motions. Traditional methods typically employ either basic rigging-and-skinning techniques [11], [114] or rely on physics-based simulations [152], [195], which requires intensive computations and specialized expertise to create a simulation-ready clothing mesh. In contrast, recent data-driven approaches [32], [40], [68], [83], [108], [111], [119], [121], [174], [224], [244], [256], either using implicit or explicit representations, have yielded promising results in this field of research.

Notably, multiple studies [111], [118], [119], [121], [244] have demonstrated the efficacy of point-based representation of clothing shapes, attributed to the compactness and topological flexibility of point-clouds. Despite the encourag-

ing achievements, there remain unresolved challenges in this field of study. The first challenge resides in the dynamics modeling of clothed humans, where the clothing deformations are supposed to be natural and smooth, both spatially and temporally, when a person performs various motions. However, existing learning-based approaches [32], [111], [118], [119], [121], [174], [244] focus on the clothing deformations associated with a single pose only, overlooking the underlying correlation and continuity of clothing deformations across a motion sequence. The second challenge relates to the progressive modeling of clothed humans, a process that mirrors the iterative refinement typically seen in artifact creation. Earlier works either model clothed humans in a single step [121], [174], [244], or employ a two-step coarse-to-fine strategy [111], [119], thereby missing the opportunity to fully exploit the benefits of progressive refinement in modeling clothes. The third challenge lies in the diversified modeling of clothed humans, which is in accordance with the real-world observation that identical outfits and motions can yield varying patterns of cloth wrinkles. Existing methods [111], [119], [121], [174], [244] are mostly deterministic, thereby limiting the range of variations in response to specific outfits and motions.

To address these challenges, we propose ClothDiffuse in Chapter 7, a diffusion-based method that learns the dynamics of clothing deformations for the realistic generation of clothing details in target motion animations. Our key insight is to involve all three significant aspects in our framework: dynamics modeling, progressive modeling, and diversified modeling of clothed humans.

1.4 Summary of Contributions

As new types of cameras and imaging modalities emerge, their utility remains largely underexplored, especially in the domain of human pose estimation and shape modeling. The unique features of these emerging technologies often provide new perspectives and solutions in this field of study. Therefore, our thesis delves into the realm of human pose estimation and shape modeling from and particularly beyond RGB cameras to explore the potential of these

emerging cameras. Specifically, our research in this thesis is structured around three pivotal components: the exploration of new cameras, the development of novel approaches, and the creation of large-scale multi-modality datasets for human pose estimation and shape modeling.

The contributions of this thesis are summarized as follows:

- **Chapter 3** (Skeletal Human Poses): Compared with single-person pose estimation, multi-person pose estimation, tracking and motion forecasting from RGB video snippets are usually more practical in real-world applications, but with more challenging cases due to the intra-frame occlusion between multiple persons. To address this, we introduce a unified framework that employs innovative spatiotemporal deformable attention module to encode the spatiotemporal relationships between images, which overcomes the intra-frame occlusion problem with RGB cameras for simultaneous implementation of three tasks in a single framework.

- **Chapter 4** (Parametric Human Shapes): Event cameras, inspired by biological vision systems, present new potential for energy-efficient 3D human pose estimation or tracking. Unlike traditional sensors that capture static images, event cameras record changes in pixel intensity, making them well-suited for capturing motion dynamics. To address this, we introduce a two-stage deep learning approach that estimates human pose and shape primarily using event data, while requiring only the first gray-scale frame instead of the entire gray-scale video. We also introduce MMHPSD, a new dataset that is the first of its kind to be publicly available. It stands as the largest event-based dataset for 3D human pose and shape estimation.

- **Chapter 5** (Parametric Human Shapes): Our subsequent event-based work further eliminates the need for the first gray-scale frame or the entire gray-scale video as input, and proposes a dedicated end-to-end sparse deep learning approach based on Spiking Neural Networks (SNNs) with

a novel spiking spatiotemporal transformer, enabling low power consumption for human pose tracking. Additionally, a large-scale synthetic dataset, SynEventHPD, is constructed for the task of event-based 3D human pose tracking. It consists of synthesized events data from multiple motion capture datasets and consequently covers a variety of motions with a total size of 45.72 hours event streams – more than 10 times larger than MMHPSD.

- **Chapter 6** (Clothed Human Shapes): Polarization cameras are known to preserve detailed surface normal maps following the physical laws of light polarization. Motivated by this physical fact, we propose a dedicated two-stage framework that leverages surface cues from polarized images for human pose estimation and shape reconstruction. A dedicated dataset, PHSPD, has been created. It consists of ∼527K frames along with corresponding pose and shape annotations. Overall there are 21 different subjects performing 31 unique actions, and ∼9.5 hours of videos are recorded in total. Empirical evaluations on the synthetic SURREAL dataset [199], as well as our real-world PHSPD dataset, demonstrate the effectiveness and applicability of our approach, showcasing that for 3D human poses and shapes estimation, 2D polarization camera could be a viable alternative to conventional RGB cameras.

- **Chapter 7** (Clothed Human Shape Animation): Depth sensors, known for their ability to intuitively, efficiently, and generally represent 3D objects, have emerged as a straightforward tool to capture 3D point-clouds. Meanwhile, going beyond static human shape reconstruction, our research focuses on the complex task of animating clothed humans with natural clothing deformations, leveraging point-cloud sequences that provide valuable geometric insights into the structure of the clothing. Existing data-driven approaches often overlook three crucial aspects in clothed human modeling: dynamics modeling, progressive modeling and diversified modeling. To tackle these challenges, we introduce ClothDiffuse, a diffusion-based method that seamlessly integrates these three crucial

aspects into clothed human modeling.

- We perform comprehensive experiments in each project to assess the effectiveness of both emerging camera technologies and innovative approaches. Our findings demonstrate that beyond traditional RGB cameras, alternative emerging sensors like event cameras, polarization cameras, and point-clouds offer promising avenues for advancements in human pose estimation and shape modeling.

Chapter 2

Literature Review

2.1 Related Research Topics

2.1.1 Human Pose Estimation

In the past few years, 3D human pose estimation from single images, mainly based on RGB or depth images, has been extensively studied. Many early efforts [2], [27], [207], [259] utilize dictionary-based learning strategies to capture prior knowledge from large motion-capture datasets. Recent efforts focus on end-to-end deep learning based methods, including CNNs [104], [150] and Graph CNNs [22], [39] to estimate 3D human poses. In particular, a common framework has been adopted by a number of recent works [70], [124], [155], [196], [205], [209], [232], [252], [260], which first infer 2D poses (either 2D joint positions or heatmaps) and then lift those poses to 3D. Self-supervised learning [70], [209] and adversarial learning [205], [232] are also considered to exploit the benefits of additional re-projection or adversarial constraints.

Going beyond pose estimation, the availability of parametric human shape models, such as SMPL model [116], has fueled growing attention in single image-based human shape estimation. SMPL is a statistical low-dimensional representation of human shape, realized by principal component analysis of empirical body shapes of naked and minimally dressed humans. Early efforts focus on optimization-based methods to fit SMPL model to point-clouds or annotated pose [10], [19], [47], [48]. Recent deep learning based methods [91], [147], [199] instead learn to predict SMPL parameters under various

constraints such as 2D/3D pose, silhouette, and adversarial examples. Human body pixel-to-surface correspondence maps are also considered in [228] for parametric shape estimation. In [99], optimization and regression are integrated to form a self-improvement loop. There are also a number of recent efforts to exploit temporal information in inferring human poses and shapes from videos, including temporal constraints [93], [98], dynamic cameras [239] or event signals [227], [266]. Sensing modalities other than the classical RGB images have also been explored for human pose estimation, including polarization image [270], IMUs [202] and head-mounted devices [250].

2.1.2 Human Shape Modeling

In terms of human shape modeling, volume-based methods [172], [198], [233], [257], [258] are popular in reconstructing detailed body shapes. They unfortunately suffer from the limitation of computation scalability and lack of reliable 3D cues. The recent work, PaMIR [257], combines the parametric body model with the free-form deep implicit function to reconstruct human shapes. Unlike surface normal, the implicit field cannot provide explicit cues of human body and leads to inaccurate reconstructed shapes. Saito et al. [172], [173] introduce a pixel-aligned implicit surface function to encode detailed body surface. However, these two works are not able to handle complex poses, partly owing to the lack of complex poses in their training set. A closely related work is [264], which considers a hierarchical framework to incorporate robust parametric shape estimation and flexible 3D shape deformation. It, however, employs a network trained on additional small dataset to infer shading information, which are inherently unreliable given the lack of ground-truth information of surface normal, albedo and environmental lighting. Another related work is [192], which iteratively integrates rough depth map and the estimated surface normal for improved clothes details.

Clothed human modeling aims to learn clothing deformation dynamics from a set of 3D point-clouds or meshes of clothed human body, facilitating the generation of naturalistic clothing details in target motion animations. Existing methods can be divided into two main categories based on their implicit and

explicit representation of humans:

(i) Implicit representation commonly defines surfaces as a level set of scalar implicit function. Powered by multi-layer perceptrons (MLPs), this function learns to predict the occupancy value for any given 3D position in continuous camera space for shape reconstruction [34], [37], [38], [130], [149]. Instead of relying on pre-defined templates, implicit surfaces are topologically free and are able to model diverse complex topology. This promising technique is also widely applied in human reconstruction [49], [82], [148], [172], [173], [178], [257] and clothes modeling [17], [31], [32], [40], [108], [132], [160], [174], [194], [213], [224], [256]. Most of these methods [31], [32], [40], [49], [108], [132], [174], [194], [213], [224] adapt the pipeline of reconstructing clothed human in canonical space first and then animating with predicted skinning weights and pose-aware clothing deformation, while some others [17], [82], [148], [160] follow the part-based modeling of clothed humans. However, implicit human modeling is known to require a cubic increase in both time and computation with respect to the resolution of discretization for reconstructing explicit surfaces [34], [111], [121], [244].

(ii) Explicit representation is another popular stream that assumes a specific mesh-based template is provided or predicted for human clothes modeling [16], [18], [21], [68], [83], [94], [141], [152], [221]. Although these template-based approaches have shown their robustness and efficiency in clothes modeling, they are still limited to the generalization ability of various typologies and also the requirement of registration or canonization to raw scans. Following efforts [120], [175], [180] attempt to alleviate this limitation by using a generative model to produce template mesh of flexible topology. They still suffer from the expressiveness of a specific template. On the contrary, point-based representation supports both compactness and arbitrary topology. Earlier works [1], [53], [109] generate sparse point set for 3D object reconstruction, and following works [12], [45], [66] group point-clouds into structured patches with each patch representing a 2D UV map, allowing dense sampling of points on the patch to model detailed surface geometry. SCALE [118] is the first work that successfully applies to human clothes modeling and the following effort,

POP [121], further demonstrates the ability of a single model for arbitrary clothes types. To address the issue of varying topology of clothes, FITE [111] proposes to learn an implicit model to reconstruct a coarse template of clothes and then add explicit pose-dependent deformation. A similar idea is also employed in SkiRT [119] that introduces a coarse-to-fine process. The most recent work, CloSET [244], suggests to learn pose features on a body surface to tackle the discontinuity of the UV map used in [118], [121].

2.1.3 Multi-Person Pose Estimation and Tracking

Multi-person pose estimation from monocular images has been extensively investigated in the past few years. Existing methods can be divided into three categories: bottom-up [24], [51], [52], [67], [103], [125], [158], [241], top-down [33], [35], [36], [64], [137], [167], [197], [206], [215] and single-stage [14], [86], [127], [145], [181], [189], [214], [217].

Bottom-up methods detect 2D joints first and estimate 2D or 3D pose with different association approaches, such as integer linear program [158], [241] or part affinity fields [24]. In addition, Gu et al. [67] formulate 3D pose estimation as a Perspective-N-Point optimization problem based on detected multi-person 2D poses via [24] and shows good performance with high efficiency for 3D pose estimation. Recently, Fabbri et al. [51] extends 2D heatmaps to 3D compressed volume for 3D joints detection for multiple persons. There is also effort [125] using depth map for efficient multi-person 2D pose estimation with CNNs and knowledge distillation at multiple architecture levels. A most recent work [103] trains a Hourglass model [142] to predict multi-person 2D keypoints heatmap with peak regularization and employs greedy keypoint association to obtain multi-person 2D poses.

Top-down methods first detect the person bounding box and then apply single person pose estimation on the cropped region. A pose proposal generator is employed in [168] followed by a pose refinement regressor. RootNet [137] infers multi-person 3D pose by detecting absolute 3D root localization first and then estimating root-relative single-person 3D pose. Wei et al. [219] present a view-invariant hierarchical correction network on top of an initial estimated

single-person 3D pose to learn the 3D pose refinement under consistent views, and then use a view-invariant discriminative network to enforce high-level constraints over body configurations. HMOR [206] encodes interaction information of multiple persons as the ordinal relations of depths and angles hierarchically. The recent effort [35] applies graph and temporal convolutional neural networks for multi-person pose estimation in a video. In general, top-down methods estimate more accurate poses than bottom-up counterparts, but with the expense of more computation. Integrating bottom-up and top-down is considered in [36] to complement each other. Multi-view top-down approaches are investigated in [167], [197], where humans are detected and integrated from multi-view sources in a 3D volume and then regressed to estimate 3D poses. A most recent effort [215] proposes knowledge transfer network to learn the 2D-3D correspondences for multi-person 3D dense pose estimation because of insufficient and imbalanced 3D labels.

Single-stage methods are emerging in recent years for both pose estimation [14], [15], [86], [127], [145], [181], [217], [254] and parametric human shape estimation [189]. They achieve multi-person pose estimation in a single stage, wherein the entire inference process is conducted in a single forward pass using a unified end-to-end model. These approaches eliminate the need for person detection as required in top-down methods and joints association typical in bottom-up methods.

Multi-person pose tracking aims to track multi-person poses in a video. Recently, [191] provides a survey of multiple pedestrian tracking based on the tracking-by-detection framework. For 2D pose tracking, Girdhar et al. [64] use top-down approaches to estimate frame-based multi-person poses and then link predictions over time using bipartite matching. [79], [222] employ a similar top-down detection strategy to achieve multi-person pose estimation, but rely on a flow-based similarity to perform tracking. Wang et al. [211] extend HRNet [187] with temporal convolutions and show impressive joint pose estimation and tracking. In contrast, Raaj et al. [161] propose an efficient bottom-up approach by extending the spatial affinity fields to spatiotemporal affinity fields in an RNN model. For 3D pose tracking, multi-stage approaches

are proposed in [127], [240] where per-frame multi-person 3D pose estimation is followed by a temporal constraint optimization or fitting step. In contrast, [52], [167] aggregate temporal information within the model to estimate the multi-person pose trajectory. The most recent top-down works [162], [163] achieve tracking by using 3D representation of people or predicting the future state of the tracklet, including 3D location, appearance, and pose. Besides, there are also efforts [251] exploring pose tracking with cross-view correspondence for occlusion-aware 3D tracking.

2.1.4 Shape from Polarization (SfP)

SfP focuses on inferring an object's shape (normally represented as surface normal) from a polarization image, where each channel captures the polarimetric information of the reflected light under a linear polarizer at a different angle. There are two main issues involved in SfP: angle ambiguity and the discrimination of specular and diffuse reflection. Previous efforts are mainly physics-based, which rely on additional information or assumptions to elucidate the possible ambiguities, such as smooth object surfaces [8], coarse depth map [89], [231] and multi-view geometric constraints [29], [41]. The first deep learning based method is conceived in [9] that integrates physical priors (ambiguous normal maps) with deep models for estimating normal maps. It has shown that deep models can learn to leverage the angle ambiguity and environmental noise. A follow-up work [269] advocates to first classify each pixel into different types of ambiguous angles and then obtain a fused normal map, which is shown to extract more explicit geometric cues from polarization images.

2.1.5 Event Camera and Its Applications

Event cameras [59], as a new bio-inspired technology of silicon retinas, differ notably from the conventional frame-based imaging sensors, such as RGB or Time-of-Flight cameras, including but not limited to its asynchronous and independent *address-event* representation. The output of event cameras is a sequence of "events" or "spikes". Consider the binary polarity status p that represents either brightness increase or decrease. Each readout event can be

represented as a tuple $(\mathbf{x}, t, p)$, where the event corresponds to a change in brightness at pixel position $\mathbf{x}$ (referred to as the 'address') that surpasses a predefined threshold at time t. Instead of densely capturing pixel value at a fixed frame rate for frame-based cameras, event cameras record the intensity change for each of the pixels asynchronously and independently, in case a motion occurs. Hence the temporal resolution of event cameras is much higher than conventional frame-based cameras. Moreover, as its output consisting of a spatially much sparser stream of events, event cameras typically consume considerably less energy in operation.

Event-based vision applications have witnessed a substantial increase in recent years, including camera pose estimation [60], feature tracking [63], optical flow [71], [263], multi-view stereo [166], [248], hand gesture recognition [3] and pose estimation [170], motion deblurring [85], [188], image restoration and super-resolution [210], image classification [57], [58], object recognition [95] and tracking [135], [245], [246], semantic segmentation [190], events from/to video [61], [62], depth estimation [247], among others.

As for the task of event-based human pose estimation, DHP19 [23] is perhaps the first effort in engaging CNNs models for event camera based human pose estimation. EventCap [227] aims to capture 3D motions from both events and gray-scale images provided by event cameras. This work starts with a pre-trained CNN-based 3D pose estimation module that takes a sequence of low-frequency gray-scale images as input. The estimated poses serve as the initial state and are used to infill intermediate poses for high-frequency motion capture, constrained by detected event trajectories from [63] and silhouette information gathered from the events. These methods, however, require full access to the corresponding gray-scale images as co-input. EventHPE [266] reduces this demand by the milder need of only a single gray-scale image of the starting pose. To do this, a dedicated CNNs module is trained to infer optical flow by self-supervised learning, which is used alongside with the input event stream to track 3D parametric human shapes.

2.2 Related Model Architectures

2.2.1 Spiking Neural Networks (SNNs)

SNNs have been an emerging learning framework in recent years. Spiking neuron, the basic element in SNNs, works by imitating the transmitting mechanism in mammalians' visual cortex [59]. A spiking neuron maintains a membrane potential, which could be changed only when spikes (*events*) are received from its connected preceding neurons. A spike is produced when the neuronal potential exceeds a preset threshold. Different from the neuron in traditional artificial neural networks (ANNs), no output would be produced by spiking neurons as long as their potentials are below spiking threshold, thus no computation takes place – the root cause of the remarkable efficiency and sparsity of SNNs when comparing to the dense and computational-heavy ANNs.

Training large-scale SNNs from scratch presents a significant challenge. To address the non-differentiable issue of neuronal spiking function, one branch of research focuses on converting trained ANNs to SNNs [44], [72], [106], [171], [229] (ANN2SNN). Typically, these methods map the non-linear activation layer in a trained ANNs to the neuron spiking layer, and then scale its threshold or the weights connected to other neurons accordingly. It is worth noting that only in the realm of classification tasks, excellent results have been demonstrated by the SNNs methods. The performance is still unclear in fine-grained regression tasks and specifically in human pose estimation. Meanwhile, another branch of research focuses on training SNNs from scratch, often by following the back-propagation through time (BPTT) framework and applying surrogate derivatives [107] to approximate the gradient of neuronal spiking function. This line of works has delivered impressive performance in classification tasks [57], [58], [107], [235], [236], [262] as well as regression tasks [71]. There have also been efforts [234], [245] in proposing mixed frameworks blending SNNs and ANNs, in order to maintain a good balance in efficiency and performance for event-based tasks.

2.2.2 Transformer

Transformer is originally proposed in [200] where self-attention is used and achieves the state-of-the-art results on many sequence-based tasks. DETR [26] and VisTR [216] are recent inspiring attempts to apply transformer in the end-to-end object detection and instance segmentation. Besides, multi-object tracking with Transformer has also been investigated in [129], [242]. However, due to the high computational requirement of the dot-product attention, both DETR and VisTR can only process low-resolution feature maps, which limit their accuracy. Deformable attention mechanism is proposed in [265] to tackle this issue, showing strong accuracy in detecting small objects. Transformer is recently applied to single person pose estimation, where self-attention module is directly applied to the positions of joints or mesh vertices [105], [110], [255], and also multi-frame image features [204], [214]. The most recent work [181] has also employed transformer for multi-person pose estimation, where the query-based self-attention transformer is used to regress multi-person poses for a single frame.

Spiking Transformer has emerged very recently as a new SNNs architecture. To avoid confusion, it is important to clarify that the spiking transformers presented in [245], [247] are not SNN-based transformers, but rather ANN-based or mixed models. The two recent works [235], [262] are most related to our proposed spiking spatiotemporal transformer. In MA-SNN [235], multi-dimensional attention is proposed in an SNNs framework, yet this attention is instead based on real values of membrane potentials, thus in a sense violating the efficiency design of SNNs. The most recent work, Spikformer [262], proposes to remove softmax function to accelerate the computation of self-attention on binary spike tensors. But a scaled dot-product is directly adopted to compute the similarity score of binary spike vectors.

2.2.3 Diffusion Generative Models

Diffusion model is a paradigm of generative neural models based on the stochastic diffusion process. Originating from Thermodynamics [184], the diffusion

process gradually adds a small amount of Gaussian noise to a sample from the data distribution. The reverse process is to recreate the true sample from a Gaussian noise, where a neural model is learned to gradually denoise the sample. Finally, sampling from the learned data distribution is done by denoising from Gaussian noise. Earlier works [75], [186] successfully apply diffusion models in image generation. Following works [46], [76], [143], [164], [169] further scale up the generation resolution and show the superiority of diffusion models in many generation tasks, including class and text-conditioned image generation, image super-resolution and inpainting. There are also recent attempts to employ diffusion models in 3D representation, such as shape generation and completion [243], 3D object reconstruction [226], text-to-3D generation [159] and also human shape reconstruction [179].

Regarding the generative models in clothed human shape modeling, SMPLicit [40] proposes to learn a latent representation of body shapes and garments for the generation of clothed humans in 3D space. Similarly, gDNA [31] learns latent codes to generate detailed 3D canonical shapes of people in a variety of garments with corresponding skinning weights. A concurrent work [179] introduces diffusion models into the iterative stereo matching network for high-quality human shape reconstruction.

2.3 Related Datasets

A number of human pose and shape datasets have been released in recent years, including MPII [2], MS COCO [112] and PoseTrack [5], which provide in-the-wild RGB images and 2D pose annotations. Human3.6M [80], MPI-INF-3DHP [126] and 3DPW [123] are three benchmark datasets with 3D annotations for human pose estimation from RGB images. There are also human shape datasets [172], [192], [258] that mainly consist of RGB-Depth images and the corresponding human shape annotations. However, these existing datasets are limited to RGB and/or depth images, constraining their utility in some projects in this thesis that investigate the application of new cameras or modalities for human pose estimation and shape modeling. We

summarize a tally of widely-used RGB-based human pose and shape datasets in Tab. 2.1, compared in terms of number of subjects (Sub), number of actions (Act), multi-modality dataset or not (MM), annotated poses (Pose) and shapes (Shape). Compared with other datasets in Tab. 2.1, our PHSPD dataset, denoted as a multi-modality dataset, encompasses not only RGB images but also includes depth, polarization images and/or event streams. A few visual examples are shown in Fig. 2.1.

Dataset	Sub	Act	MM	RGB	Depth	Pose	Shapse
MS COCO [112]	-	-	✗	330K	✗	2D	✗
MPII [2]	-	-	✗	40K	✗	2D	✗
PoseTrack [5]	-	-	✗	22K	✗	2D	✗
MPI-3DHP [126]	8	-	✗	1.3M	✗	3D	✗
3DPW [123]	7	8	✗	51K	✗	3D	✓
Human3.6M [80]	11	15	✗	3.6M	0.45M	3D	✗
PHSPD (ours)	21	31	✓	2.1M	2.1M	3D	✓

Table 2.1: A tally of widely-used datasets for human pose and shape estimation.

Additionally, event-based datasets are crucial for data-driven approaches to attain their satisfactory performance. This has motivated a variety of event-based datasets released in recent years, including DvsGesture [3] for hand gesture recognition, CIFAR10-DVS [102] and ES-ImageNet [113] for object classification, DSEC-Semantic [190] for semantic segmentation and EED [135] and FE108 [246] for object tracking. Unfortunately, existing benchmark datasets [43], [80] are mostly based on conventional RGB or depth cameras for human pose estimation, thus are infeasible to be directly used in event-based tasks, given the fundamental differences between event and conventional cameras. DHP19 dataset [23] is the earliest one but has limited amount of events data and lacks pose variety. Therefore, we create our own in-house multi-modality dataset, MMHPSD [266]. To our knowledge, MMHPSD is the largest real event-based 3D human pose and shape dataset, and is the first publicly available dataset of such type.

Furthermore, although MMHPSD dataset provides more than 4.5 hours event stream and 21 different types of action, it still lacks pose variety because of in-house constrained environment. We further augment MMHPSD dataset

Dataset	R/S	Sub	Str	Len (hrs)	AvgLen (mins)	Pose
DHP19 [23]	Real	17	33	0.80	1.46	✓
EventCap [227]	Real	2	6	-	-	✓
MMHPSD [266]	Real	15	178	4.39	1.48	✓
EventH36M	Syn	7	835	12.46	0.90	✓
EventAMASS	Syn	13	8028	23.54	0.18	✓
EventPHSPD	Syn	12	156	5.33	2.05	✓
SynMMHPSD	Syn	15	178	4.39	1.48	✓
SynEventHPD (Total)	Syn	47	9197	45.72	0.30	✓

Table 2.2: A tally of event-based datasets for 3D human pose estimation and tracking.

by synthesizing events data from multiple human motion capture datasets (*i.e., Human3.6M [80], AMASS [122], PHSPD [270] and MMHPSD-RGB [266]*), and finally provide a large-scale synthetic dataset, SynEventHPD, with a rich variety of poses for event-based human pose tracking in [267]. We present a tally of event-based datasets for human pose estimation and tracking in Tab. 2.2, compared in terms of real or synthetic data (R/S), number of subjects (Sub), number of event streams (Str), total time length of all the event streams in hours (Len), average time length of each stream in minutes (AvgLen), annotated poses (Pose). Visual examples of our datasets are provided in Fig. 2.2.

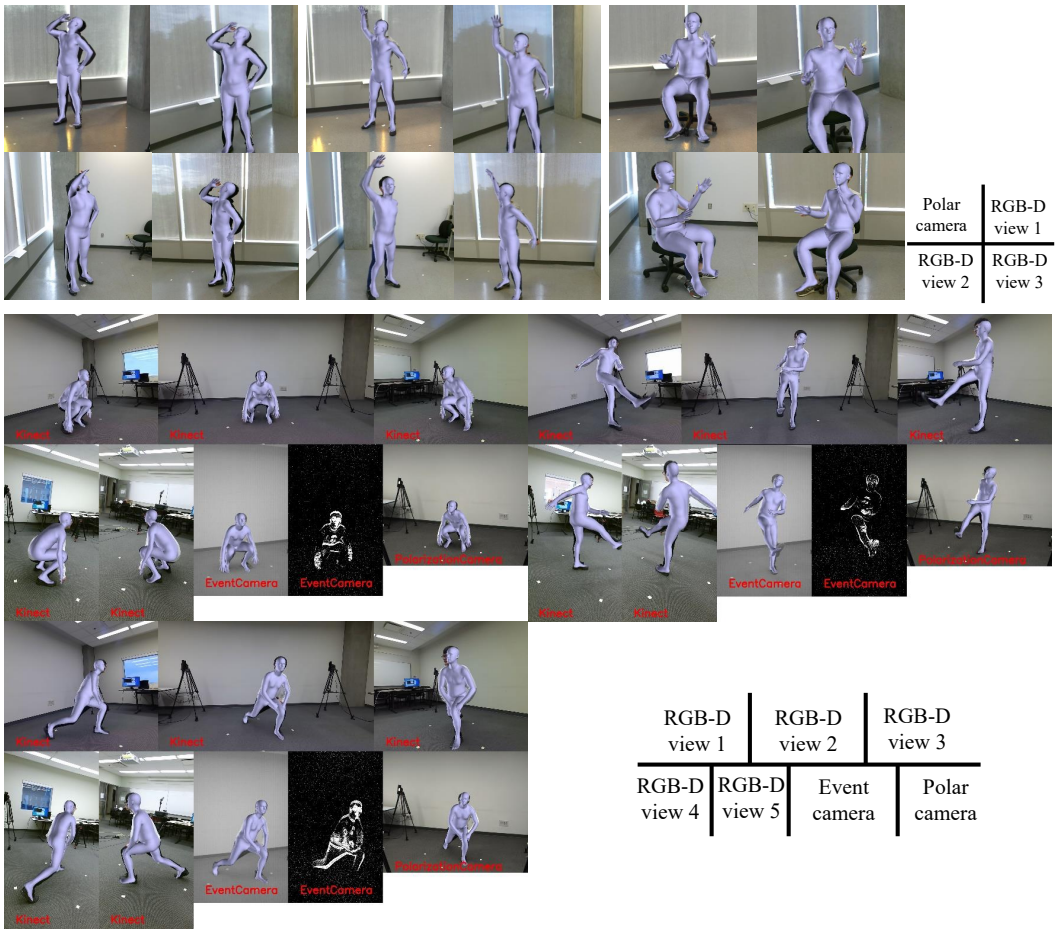


Figure 2.1: Exemplar multi-view figures with annotated shape and pose of our PHSPD dataset.

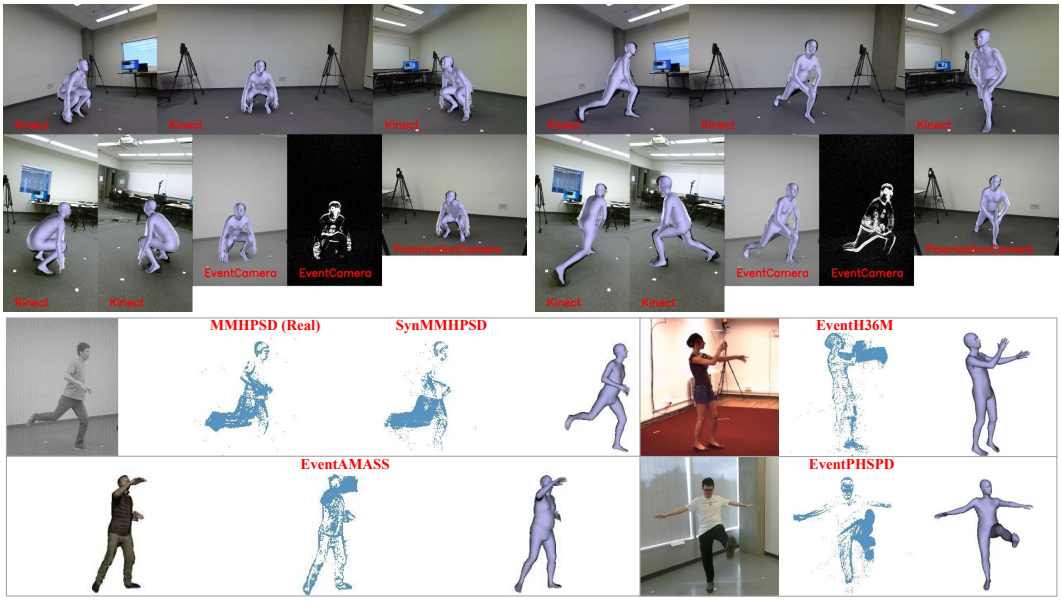


Figure 2.2: Exemplar multi-view figures with annotated shape and pose of our MMHPSD dataset.

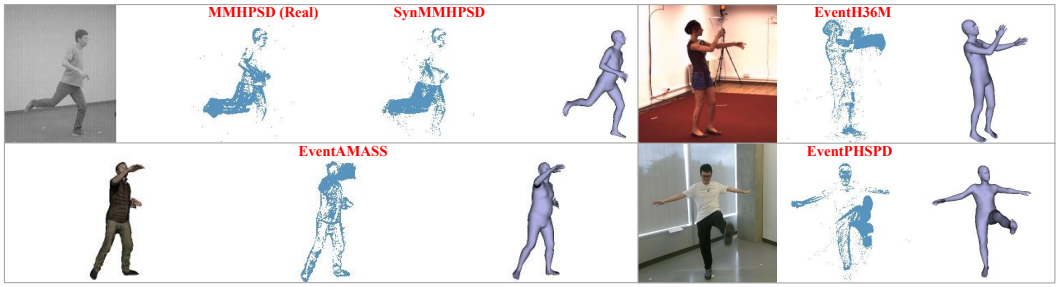


Figure 2.3: Exemplar figures with annotated shape and pose of our synthetic event-based SynEventHPD dataset.

Chapter 3

Multi-Person 3D Pose Estimation Tracking and Forecasting on an RGB Video Snippet

3.1 Introduction

Understanding multi-person 3D pose from RGB videos is a crucial research area in computer vision. This problem primarily comprises three complex tasks: multi-person pose estimation, tracking, and motion forecasting. These three tasks have broad applications, including but not limited to human action recognition, behavior analysis, pedestrian tracking, re-identification, human-computer interaction, and video surveillance [13], [138], [185], [191]. For instance, in the context of analyzing human behavior in crowded settings, multi-person pose estimation and tracking are instrumental in generating accurate behavioral data. Similarly, motion forecasting aids in predicting behavior and refining future areas of interest for the system.

Prior research has generally focused on either individual tasks [14], [24], [67], [103], [125], [191], [197], [215], [219], [270] or employed multi-stage methods that address multiple tasks independently [25], [52], [162], [167], [222]. However, these approaches often fail in complex scenarios featuring significant occlusions, as shown in Fig. 3.1. Such limitations can adversely affect both pose estimation and individual tracking in videos, leading to unreliable motion

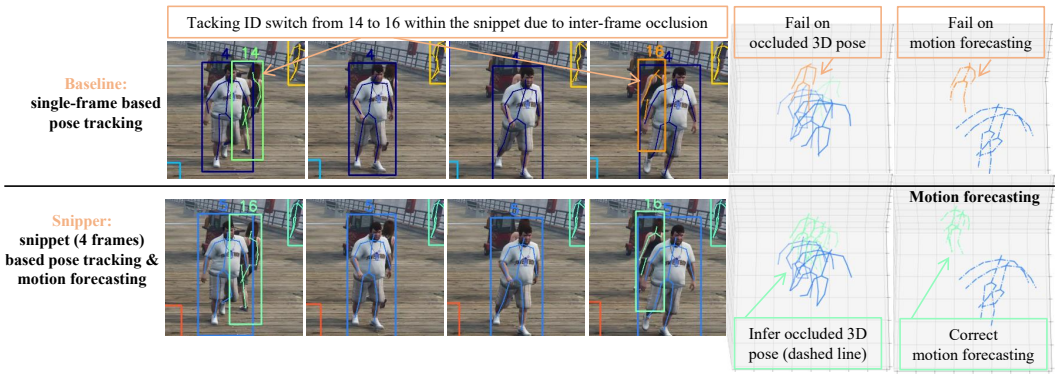


Figure 3.1: A practical yet challenging example to track multi-person pose and motion forecasting.

forecasting due to incorrect historical data.

Three primary shortcomings contribute to these failures: 1) Single-task methods usually neglect temporal information, which is especially evident in single-frame-based multi-person pose estimation [24], [67], [103], [125], [215]. 2) Multi-stage approaches often make sub-optimal decisions by treating the tasks separately, without unified reasoning. For example, previous works [25], [69], [201], [203] typically consider pose tracking and motion forecasting as disconnected modules. 3) The existing methods do not adequately leverage the interrelationships among the three tasks. Intuitively, accurate multi-person pose estimation can facilitate more robust tracking. In turn, robust tracking enriches the context for pose estimation and generates the essential historical data needed for reliable motion prediction.

To overcome these challenges, we introduce Snipper, a unified framework designed to concurrently estimate, track, and forecast multi-person 3D poses across a sequence of contiguous RGB frames. Unlike traditional approaches, Snipper performs these three tasks in a unified space, outputting 2.5D multi-person pose trajectories along with future motion predictions from a video snippet in a single processing stage. The framework draws inspiration from the query-based DETR framework for object detection [26], [216], [265]. There is also a recent effort proposing TubeDETR [230] for spatio-temporal video grounding, which focuses on reasoning within the realms of space, time, and language. However, Snipper distinguishes itself with a novel contribution: an

efficient and powerful spatiotemporal deformable attention module specifically tailored for fine-grained video understanding tasks.

In particular, our attention mechanism employs sparse spatial deformable attention [265] to handle high-resolution, multi-scale image features, which is crucial for better image-aligned tasks. A straightforward extension of this mechanism to video—by regressing a space-time offset and sampling directly in 3D space—presents challenges. Specifically, interpolating in the time domain becomes ambiguous without temporal correspondences, and the image features at identical spatial locations can vary across frames due to object or camera motion.

To address these issues, we propose limiting the temporal offset to predefined integer frame indices and confining the spatial offset regression to those specific frames. This approach enables the aggregation of per-frame image features across both space and time. Unlike the self-attention mechanism used in the work by Carion et al. [26], our technique is not only efficient but also preserves the spatiotemporal relationships among multi-frame and multi-scale features. This is accomplished through the deformable attention mechanism, which aggregates spatiotemporal features. Compared to the spatial attention described in [265], our strategy incorporates additional temporal considerations. This optimized approach is particularly crucial for compensating for information loss caused by occlusions or motion blur within a video snippet. Further details and comparative analysis are provided in Sec. 3.2.3 and 3.3.4, as well as in Fig. 3.11.

Leveraging our novel spatiotemporal deformable attention module, we construct a deformable transformer designed for the simultaneous execution of all three tasks: pose estimation, tracking, and motion forecasting. Specifically, an encoder, detailed in Sec. 3.2.4, first processes the multi-frame feature volume extracted by a CNN backbone. It employs our attention module to update the features of each voxel by aggregating spatiotemporal information. This enriched feature volume serves as the memory input for the transformer decoder, as described in Sec. 3.2.5. Multi-person pose queries then accumulate pose trajectory features from this memory using the same attention module.

Ultimately, these queries are used to regress and predict multi-person pose trajectories in observed frames, as well as to forecast future motions (Sec. 3.2.6). To ensure consistent tracking across an entire video, Snipper operates on overlapping snippets and correlates pose trajectories based on common frames between consecutive snippets. Importantly, this approach eliminates the need for additional appearance descriptors for tracking.

Our contributions are summarized as follows:

- We propose Snipper, a unified framework for simultaneous multi-person 3D pose estimation, tracking, and motion forecasting from a video snippet. To our knowledge, the proposed framework is the first one that jointly solves these three tasks in a single stage.
- We propose an efficient yet powerful spatiotemporal deformable attention mechanism in the transformer to aggregate spatiotemporal information from multi-scale and multi-frame feature volumes. Its effectiveness and efficiency are discussed in Sec. 3.2.3 and validated in Sec. 3.3.4, and our proposed spatiotemporal attention module is also general to other image-aligned video understanding tasks. [1].
- We validate the proposed framework on three challenging datasets: JTA [52], CMU-Panoptic Studio [88], and PoseTrack18 [4]. We show that a generic Snipper model presents competitive performance on all three tasks of pose estimation, tracking, and motion forecasting compared with specialized baselines that tackle only one or two tasks.

3.2 Method

Snipper is a unified framework that simultaneously addresses three tasks from an RGB video snippet of T observed frames: multi-person 3D pose estimation, tracking, and forecasting for the future T_f frames. An overview of our pipeline is shown in Fig. 3.2. Taking a snippet of T consecutive RGB images as the input, a CNN is used to extract per-frame image features, which are stacked into

[1] Our code is publicly available at https://github.com/JimmyZou/Snipper

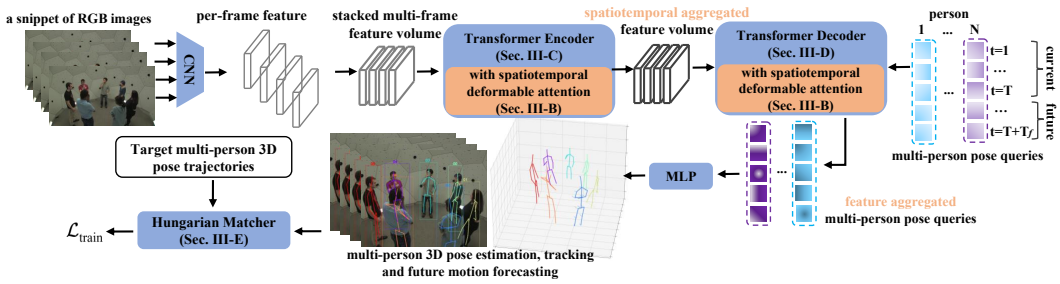


Figure 3.2: Overview of our proposed approach, Snipper.

a multi-frame feature volume. This feature volume is fed into the transformer encoder (Sec. 3.2.4) that employs a novel spatiotemporal deformable attention (Sec. 3.2.3) to aggregate features. Given the spatiotemporally aggregated feature volume from the encoder, the transformer decoder (Sec. 3.2.5) aggregates pose features from it via spatiotemporal deformable attention. Those aggregated pose features are used to update pose queries for N people of T observed frames and T_f future frames. The updated queries are regressed to estimate each person's 3D pose tracking over $T + T_f$ frames. We use Hungarian match to find an optimal permutation of a fix set of predicted 3D pose trajectories to a set of target trajectories to compute the multi-person pose losses for training (Sec. 3.2.6).

3.2.1 Preliminary

Star pose representation. We represent the 3D human pose as $\mathbf{P} = \{\mathbf{J}, \mathbf{V}, o\}$ where $\mathbf{J} = \{J_i : J_i \in \mathrm{R}^3\}_{i=1}^{N_J}$ is the set of joint locations, $\mathbf{V} = \{V_i : V_i \in [0, 1]\}_{i=1}^{N_J}$ is the joint visibility, and $o \in [0, 1]$ represents the probability of a person's occurrence. Each individual joint position J_i is modeled by the offset $J_i^{\text{offset}} = \{\Delta x, \Delta y, \Delta d\}$ from the global root $J^{\text{root}} = \{x, y, d\}$, *i.e.*, $J_i = J^{\text{root}} + J_i^{\text{offset}}$, where (x, y) are the 2D image location of the joint and d is its depth to the camera center respectively.

Depth and joint offset normalization. As the absolute root depth d depends on the camera focal length f_c, we normalize the root depth by $\tilde{d} = d/f_c$, similar to [35]. In addition, the magnitude of 2D joint offset $(\Delta x, \Delta y)$ is in pixel distance and thus depends on the depth of the person. That is, a

person's joint offset will become smaller if it moves far away from the camera, which tends to make the training unstable.

We assume the camera's intrinsic parameters are (f_c, c_x, c_y) where f_c is the focal length and (c_x, c_y) is the center of image. According to the pinhole camera model, we have $x = \frac{X}{d} \cdot f_c + c_x$ and $y = \frac{Y}{d} \cdot f_c + c_y$, where (X, Y, d) is the 3D position and (x, y) is the projected 2D position on the image. We can avoid predicting the focal length by normalizing the depth d with f_c, *i.e.*, $\tilde{d} = d/f_c$. Then for the joint offset $(\Delta x, \Delta y)$, we have $\Delta x = \frac{\Delta X}{d} \cdot f_c$ and $\Delta y = \frac{\Delta Y}{d} \cdot f_c$, which shows that the magnitude of the 2D offset in pixel distance is proportional to f_c/d. Therefore, we propose to normalize the joint offsets with the normalized depth, *i.e.*, $\Delta\tilde{x} = \Delta x \cdot \tilde{d}$ and $\Delta\tilde{y} = \Delta y \cdot \tilde{d}$. Then, $(\Delta\tilde{x}, \Delta\tilde{y})$ has the identical magnitude to the joint offset $(\Delta X, \Delta Y)$ in 3D space. Thus, the magnitude of 2D normalized joint offset only depends on the pose of the person, which is more consistent across identities.

During inference, we assume the camera's intrinsic parameters are known and that they have a fixed aspect ratio. Otherwise, we use a default focal length and pad the image to the predefined aspect ratio. Finally, a 3D joint position (X, Y, d) can be converted from the 2.5D joint representation $(x, y, \tilde{d})$.

3.2.2 Frame-Level Feature Extraction

Given an RGB video snippet of T frames, a CNN is used to extract per-frame features of size $H \times W \times C$. We stack these T frame-level features through time and obtain the multi-frame feature volume $\mathbf{F} \in \mathbb{R}^{T \times H \times W \times C}$. Note that the multi-scale pyramid features $\{\mathbf{F}^l\}$ extracted by the CNN can be easily applied in the subsequent Transformer Encoder and Decoder for fine-grained spatiotemporal feature extraction. Details are illustrated in Sec. 3.2.3 and Fig. 3.5.

3.2.3 Spatiotemporal Deformable Attention

Spatiotemporal deformable attention module is shown to produce more informative features for pose tracking from the stack of multi-frame feature volume

$\mathbf{F} \in \mathbb{R}^{T \times H \times W \times C}$ (validated in Sec. 3.3.4). Such aggregation is crucial to mitigate common issues of inter-frame information loss, such as self and partial occlusion.

We summarize our proposed spatiotemporal deformable attention in Fig. 3.3. Let $q \in \mathbb{R}^C$ be the query specified at the position $p = (x_q, y_q, t_q)$ of the multi-frame feature volume, where $x_q,\ y_q \in [0, 1]$ are the normalized pixel spatial positions and t_q is the integer frame index of the query q. Then, q is passed through two MLPs to regress 2D offsets $\Delta p_{t,k}(q)$ and corresponding attention weights $\alpha_{t,k}(q)$ normalized by the soft-max function. t is an integer specifying the temporal frame in a pre-defined set of neighboring frames $\mathbf{S}(t_q) = \{t_q - 1, t_q, t_q + 1\}$, and k indexes the offsets on each temporal frame. Note that we do not regress time offset, but only 2D spatial offsets $\Delta p_{t,k}$ on each frame of $\mathbf{S}(t_q)$. We execute this process in parallel by multiple independent heads h and form the final aggregated feature q_{final} by passing the concatenated feature from each head through a linear layer. This process is described as

$$q_{\text{final}} = \sum_h W'_h \Big[\sum_{t,k} \alpha_{t,k}(q) \cdot W_h \mathbf{F}\big(p + \Delta p_{t,k}(q)\big) \Big], \tag{3.1}$$

where W_h and W'_h are parameters of the linear layers.

Discussion. There are several alternatives to implement spatiotemporal attention with details displayed in Fig. 3.4:

- (a) *Self-attention.* Following VisTR [216], we flatten multi-frame features to a 2D matrix of shape $THW \times C$ and applies attention to all THW voxels. This attention mechanism is costly for high-resolution feature maps and also breaks the local spatiotemporal relationship for better image-aligned tasks.

- (b) *Naïve spatial deformable attention* [265]. We reshape the feature volume $\mathbf{F}$ to the shape $H \times W \times CT$, where the channel size becomes CT after concatenating multi-frame temporal features at the same image position. Then the spatial deformable attention is applied on the spatial domain $H \times W$ to aggregate spatiotemporal features. However, this

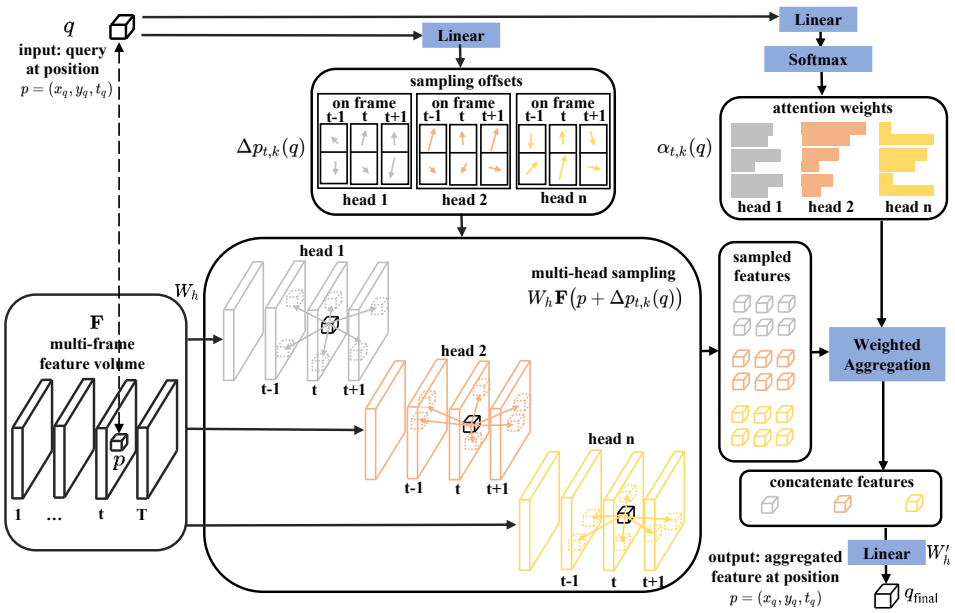


Figure 3.3: Spatiotemporal deformable attention.

naïve extension fails to consider object or camera motions within a video snippet. With these motions, image features at the same spatial position across frames often change, but the temporal features are still aggregated at fixed spatial positions.

- (c) *Direct 3D sampling.* This approach regresses space-time offsets and directly samples in the 3D space $T \times H \times W$, where the interpolation is performed in both spatial and temporal domain, *i.e.*, $t \in [1, T]$ is a fractional value instead of an integer frame index. However, the temporal interpolation is costly and ill-defined without known correspondences between frames such as optical flow, which leads to defects in the aggregated temporal features.

- (d) *Entire snippet sampling.* Another scheme is to sample on all frames of the input snippet for the query at (x_q, y_q, t_q), with offsets restricted on each temporal frame in the snippet.

- (e) *Neighboring frame sampling (ours).* Our approach limits the sampling range to be the immediate neighboring frames $\mathbf{S}(t_q)$. Despite the short temporal connection at each spatiotemporal deformable attention

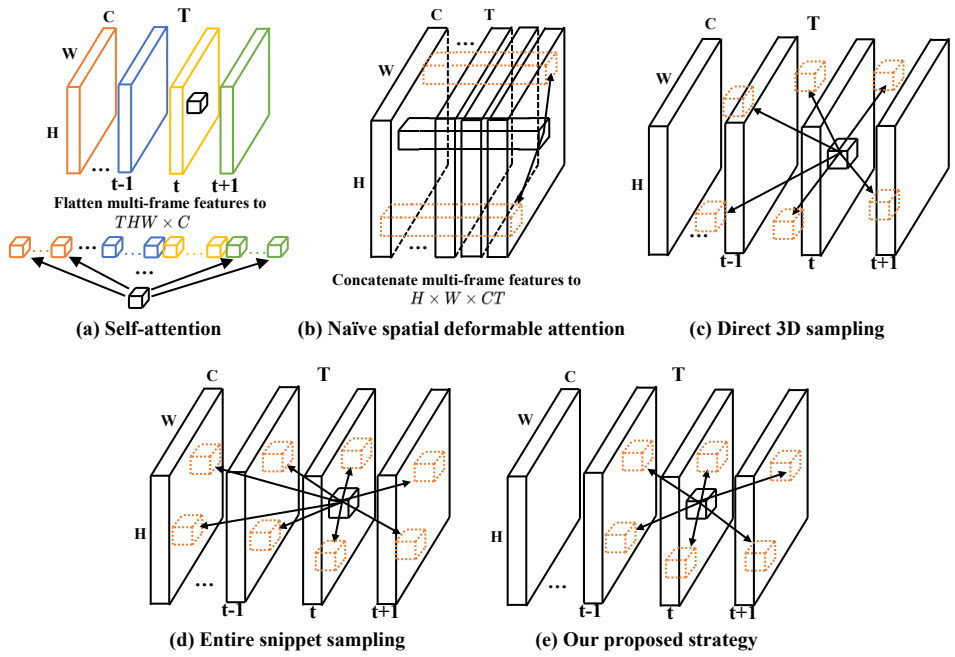


Figure 3.4: Discussion of different attention mechanisms.

module, the temporal information is still fully accumulated due to the multiple layers of the attention module in the transformer encoder. Compared with the approaches mentioned above, our proposed mechanism requires less computation, but without any performance reduction. More results and analysis are presented in Sec. 3.3.4 and Fig. 3.11 that validate the effectiveness and efficiency of our proposed strategy.

In summary, our deformable attention mechanism surpasses the conventional convolution used in CNNs in terms of flexibility for feature aggregation, as it does not rely on a fixed receptive field like traditional convolution. In contrast to self-attention, which processes the entire feature volume, our deformable attention focuses on aggregating features around the query position. This targeted approach significantly reduces feature redundancy, especially for elements far from the query position, enhancing the efficiency and effectiveness of our model.

Extension to multi-scale features. Our proposed spatiotemporal deformable attention mechanism can be naturally applied to multi-scale multi-frame features extracted by the CNN backbone. The process is summarized

in Fig. 3.5. For the query vector q at the position $p = (x_q, y_q, t_q)$, we pass it through two linear layers to regress the sampling offsets $\Delta p_{t,k,l}$ and the attention weights $\alpha_{t,k,l}$ for all scales in parallel, where l indexes the feature volume scale. We use the offsets $\Delta p_{t,k,l}$ to sample the image features at multiple scales and linearly combine these sampled features using the weights $\alpha_{t,k,l}$. This process is mathematically expressed as

$$q_{\text{final}} = \sum_{h} W'_h \Big[\sum_{t,k,l} \alpha_{t,k,l}(q) \cdot W_h \mathbf{F}^l \big(p + \Delta p_{t,k,l}(q)\big) \Big], \tag{3.2}$$

where W_h and W'_h are parameters of the linear layers.

Sampling at the higher scale focuses more on local features with a relatively shorter field of perception, while at the lower scale, it obtains more global features with a relatively broader field of perception. Unlike self-attention, which is costly when attending to the feature volume globally and is unable to extend to multi-scale features, our spatiotemporal deformable attention is efficient as it sparsely samples the image feature and approximates global attention by repeating sparse sampling through several stages.

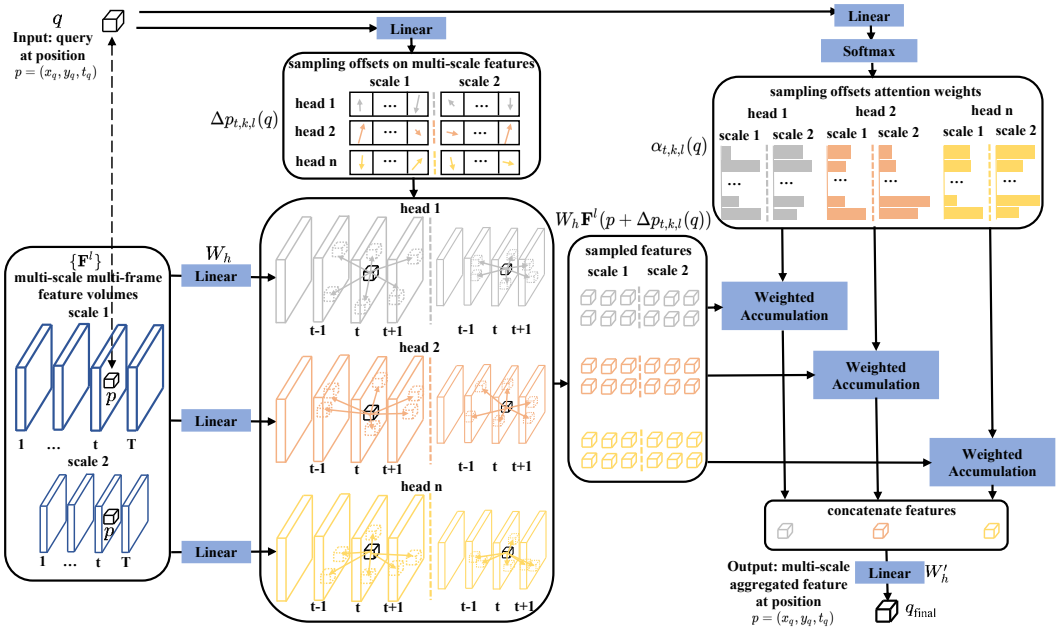


Figure 3.5: Multi-scale spatiotemporal deformable attention.

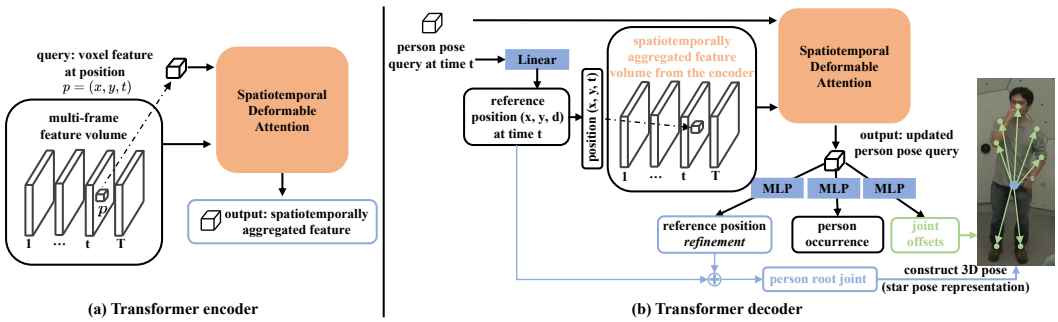


Figure 3.6: Architecture of transformer encoder and decoder with spatiotemporal deformable attention module.

3.2.4 Spatiotemporal Transformer Encoder

The goal of the transformer encoder is to generate spatiotemporally aggregated feature volume from the CNN-extracted multi-frame features. Fig. 3.6 (a) describes our transformer encoder with single layer of attention. In the encoder, for the voxel at position (x, y, t) of multi-frame feature volume $\mathbf{F}$, its voxel feature acts as the query in the attention module to aggregate spatiotemporal features from the feature volume. This process covers every voxel in the volume to create a spatiotemporally aggregated feature volume.

The encoder consists of multiple layers of the attention module, where the refined feature volume is used as input to the next layer to iteratively aggregate spatiotemporal features. In each layer, *all the voxels in all-scale feature volumes* act as the query to aggregate multi-scale spatiotemporal features. The updated feature volumes are used as input to the next layer of attention module iteratively. The process is summarized in Fig. 3.7.

Spatiotemporal positional encoding. The positional encoding of the pixel location is essential to the transformer attention mechanism. Our encoding scheme follows Wang et al. [216] for a video snippet. Each location (x, y, t) is independently encoded using $C/3$ sine and cosine functions with different frequencies, as in Vaswani et al. [200], to generate encodings. These encodings are concatenated to form the final C channel positional encoding, which is then added to the feature volume $\mathbf{F}$ and fed into the transformer encoder.

Joint heatmap supervision. Pose estimation is often better aligned with

the input image when derived from the joint heatmap or body part segmentation. [198]. Inspired by Habibie et al. [70], we enforce the first N_J channels of each temporal slice in the volume to be the multi-person joints heatmap, denoted by $\mathbf{H}_t$. Empirically, we find that this intermediate supervision improves 3D pose accuracy by 2.9% of 3D-PCK.

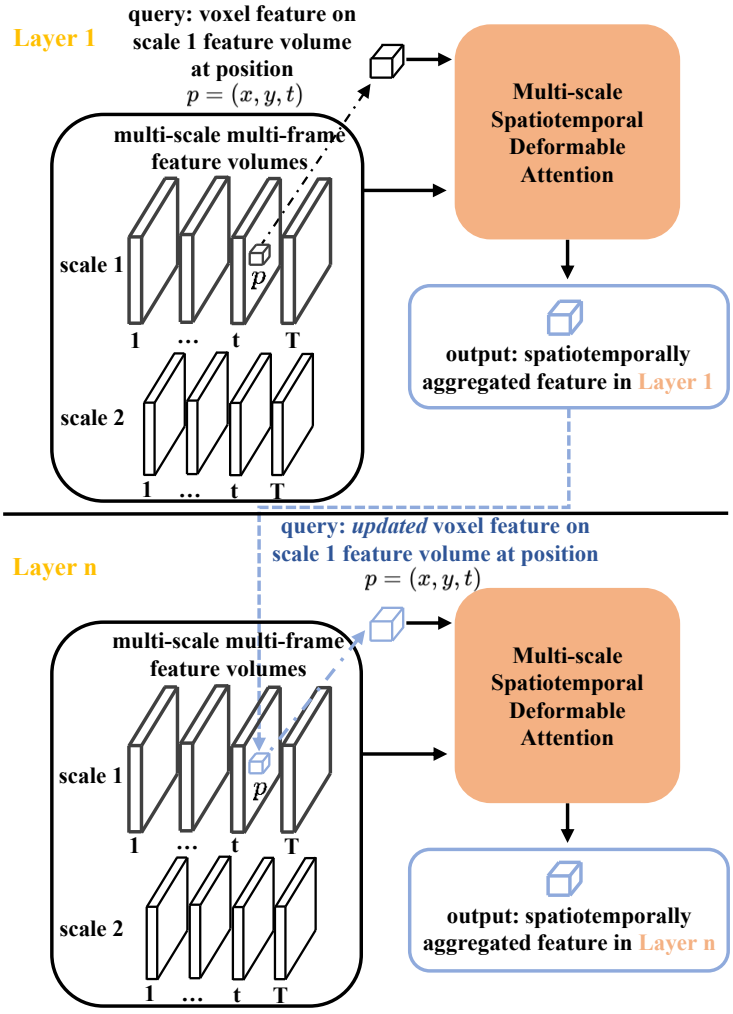


Figure 3.7: Multi-layer transformer encoder.

3.2.5 Spatiotemporal Transformer Decoder

A person pose query $q \in \mathbb{R}^C$ in the decoder is a feature vector or embedding corresponding to a single person's pose at a specific time. Given a fixed number of $N(T + T_f)$ learnable person pose queries, the decoder updates these queries by accumulating pose features from the spatiotemporally aggregated feature

volume $\mathbf{F}$. These pose queries are used to regress N people's 3D pose trajectories in a single shot. Each person's trajectory is composed of T observed poses and T_f future poses.

Temporal positional encoding. Since the pose queries of each person are agnostic about the chronological order and need to predict a pose trajectory, we add $T + T_f$ learnable temporal positional encoding to each query to make it aware of its order before feeding to the transformer decoder. We empirically observe that this helps in estimating a more accurate 3D pose trajectory.

Pose querying. The process is illustrated in Fig. 3.6 (b). In the decoder, a learnable person pose query $q_t \in \mathbb{R}^C$ at time t first regresses a reference position (x, y, d), and then conducts spatiotemporal deformable attention at the position (x, y, t) of the spatiotemporally aggregated feature volume to aggregate useful pose features. The updated person pose query passes through three MLPs to predict the refinement over reference position $(\Delta x, \Delta y, \Delta d)$, joint offsets $\mathbf{J}^{\text{offsets}}$ with joint visibility $\mathbf{V}$, and the person occurrence probability o at time t. The refined reference position is regarded as person root joint and together with joint offsets and occurrence probability, the person's 3D pose $\mathbf{P}_t$ at time t can be constructed.

Similarly, the decoder stacks multiple layers of the attention module to iteratively update the pose query. The process is illustrated in Fig. 3.8. In each attention layer, these pose queries accumulate pose features and reconstruct 3D poses iteratively. In the first layer, assuming a person pose query at time t is q_t^0, we use it to regress a reference position (x^1, y^1, d^1) and feed it into the attention module as the query to aggregate pose features at the sampling position (x^1, y^1, t) in multi-scale feature volumes. The output is the updated person pose query q_t^1 for the first layer. It is regressed to predict the joint offsets and occurrence probability to construct 3D pose $\mathbf{P}_t^1$ in the first layer, as well as position refinement $(\Delta x^1, \Delta y^1, \Delta d^1,)$ to update the reference position for the next layer. Generally, for the n-th layer, the attention module takes the updated person pose query from the last layer q_t^{n-1}, and aggregates pose features at the position $(x^n, y,^n d^n) = (x^{n-1} + \Delta x^{n-1}, y^{n-1} + \Delta y^{n-1}, d^{n-1} +$

Δd^{n-1}). The updated pose query for the n-th layer is regressed to construct the 3D pose $\mathbf{P}_t^n$ in the n-th layer.

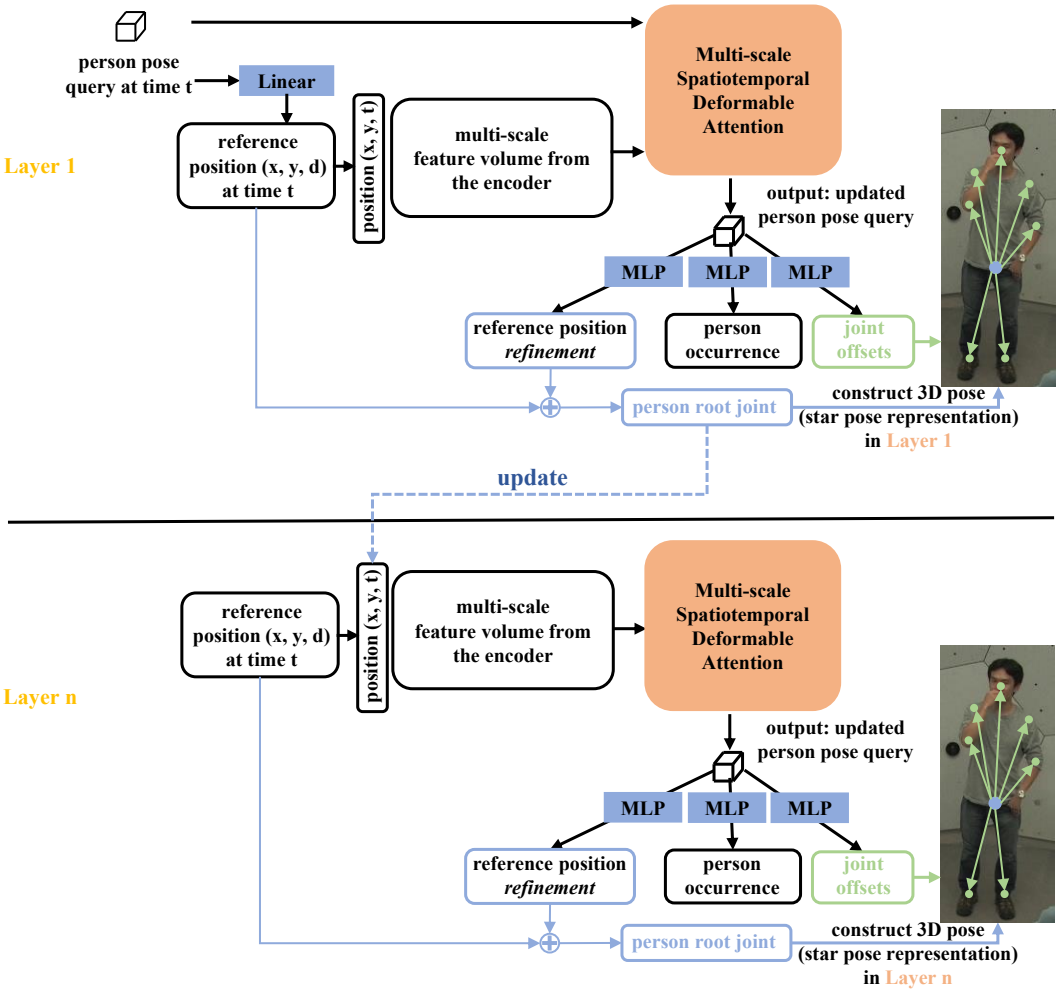


Figure 3.8: Multi-layer transformer decoder.

3.2.6 Trajectory Matching Loss

Snipper predicts a fixed number of N people's trajectories within the snippet in a single shot, where each trajectory can be represented as $\mathbf{\Gamma}_i = \{\mathbf{P}_t^{(i)}\}_{t=1}^{T+T_f}$. To supervise Snipper, we use Hungarian algorithm [100] to find the optimal matches between the predicted and target pose trajectories, and compute the pose trajectory loss for backpropagation.

Hungarian matching cost. Let $\mathbf{\Gamma} = \{\mathbf{\Gamma}_i\}_{i=1}^N$ and $\hat{\mathbf{\Gamma}} = \{\hat{\mathbf{\Gamma}}_i\}_{i=1}^M$ be the sets of predicted and target pose trajectories, respectively. We use Hungarian

algorithm to find an optimal permutation $\hat{\sigma}$ of $\mathbf{\Gamma}$ with the lowest bipartite matching cost,

$$\hat{\sigma} = \arg\min_{\sigma} \sum_{i=1}^{M} \mathcal{L}_{\text{occ}}(\mathbf{\Gamma}_{\sigma_i}, \hat{\mathbf{\Gamma}}_i) + \mathcal{L}_{\text{traj}}(\mathbf{\Gamma}_{\sigma_i}, \hat{\mathbf{\Gamma}}_i) + \mathcal{L}_{\text{vis}}(\mathbf{\Gamma}_{\sigma_i}, \hat{\mathbf{\Gamma}}_i), \tag{3.3}$$

where $\mathcal{L}_{\text{occ}}(\mathbf{\Gamma}_{\sigma_i}, \hat{\mathbf{\Gamma}}_i)$ is the negative average probability of occurrence,

$$\mathcal{L}_{\text{occ}}(\mathbf{\Gamma}_{\sigma_i}, \hat{\mathbf{\Gamma}}_i) = -\frac{\sum_t \mathbb{1}(\hat{o}_t^{(i)} \neq \emptyset) \cdot o_t^{(\sigma_i)}}{\sum_t \mathbb{1}(\hat{o}_t^{(i)} \neq \emptyset)}, \tag{3.4}$$

where $\hat{o}_t^{(i)} \neq \emptyset$ means the i-th target person occurs at time t, $\mathcal{L}_{\text{traj}}(\mathbf{\Gamma}_{\sigma_i}, \hat{\mathbf{\Gamma}}_i)$ measures the average L_1 distance between the predicted and visible target pose trajectory,

$$\mathcal{L}_{\text{traj}}(\mathbf{\Gamma}_{\sigma_i}, \hat{\mathbf{\Gamma}}_i) = \frac{\sum_k \sum_t \|\hat{V}_{k,t}^{(i)} \cdot (J_{k,t}^{(\sigma_i)} - \hat{J}_{k,t}^{(i)})\|_1}{\sum_k \sum_t \hat{V}_{k,t}^{(i)}}, \tag{3.5}$$

and $\mathcal{L}_{\text{vis}}(\mathbf{\Gamma}_{\sigma_i}, \hat{\mathbf{\Gamma}}_i)$ is the average L_2 distance between the predicted and target joint visibility,

$$\mathcal{L}_{\text{vis}}(\mathbf{\Gamma}_{\sigma_i}, \hat{\mathbf{\Gamma}}_i) = \frac{\sum_k \sum_t \|V_{k,t}^{(\sigma_i)} - \hat{V}_{k,t}^{(i)}\|_2^2}{N_j(T + T_f)}. \tag{3.6}$$

In the above cost definition, we simplify the notations with $\sum_t$ as the iteration over all the time steps $\{1, \ldots, T + T_f\}$, and $\sum_k$ as the iteration over all the joints $\{1, \ldots, N_J\}$. We follow Carion et al. [26] and adopt the detection probability instead of the log-probabilities in $\mathcal{L}_{\text{occ}}(\mathbf{\Gamma}_{\sigma_i}, \hat{\mathbf{\Gamma}}_i)$. We have observed improved matching behavior between the predicted and target pose trajectories, especially in the earlier epochs, with this strategy.

Training loss. Given the optimal permutation $\hat{\sigma}$, the matched predictions are used to compute both person's occurrence and 3D pose losses. The remaining unmatched predictions are only used to compute person occurrence loss. We define the total training loss as

$$\begin{aligned} \mathcal{L}_{\text{train}} = \sum_{i=1}^{M} \Big(& \mathcal{L}'_{\text{occ}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i) + \mathcal{L}_{\text{traj}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i) + \mathcal{L}_{\text{vis}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i) \\ & + \mathcal{L}_{\text{offset}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i) + \mathcal{L}_{\text{smooth}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i) \Big) + \mathcal{L}_{\text{heatmap}}(\mathbf{H}, \hat{\mathbf{H}}), \end{aligned} \tag{3.7}$$

where $\mathcal{L}'_{\text{occ}}$ is the negative log-likelihood for person occurrence prediction,

$$\mathcal{L}'_{\text{occ}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i) = -\sum_t \mathbb{1}(\hat{o}_t^{(i)} \neq \emptyset) \cdot \log o_t^{(\hat{\sigma}_i)}. \tag{3.8}$$

$\mathcal{L}_{\text{traj}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i)$ and $\mathcal{L}_{\text{vis}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i)$ are defined in Eq. (3.5) and (3.6) with the permutation replaced with the optimal one $\hat{\sigma}_i$ for the computation of training loss.

For the following losses, we drop the superscript $\hat{\sigma}_i$ and i from J^{offset} and $\hat{J}^{\text{offset}}$ for simplicity. $\mathcal{L}_{\text{offset}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i)$ measures the average L_1 distance between the predicted and target visible joint offsets for the supervision of a single person's pose,

$$\mathcal{L}_{\text{offset}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i) = \frac{\sum_k \sum_t \|\hat{V}_{k,t} \cdot (J^{\text{offset}}_{k,t} - \hat{J}^{\text{offset}}_{k,t})\|_1}{\sum_k \sum_t \hat{V}_{k,t}}, \tag{3.9}$$

$\mathcal{L}_{\text{smooth}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i)$ is the average L_2 smoothness of predicted joint offsets between frames within a video snippet,

$$\mathcal{L}_{\text{smooth}}(\mathbf{\Gamma}_{\hat{\sigma}_i}, \hat{\mathbf{\Gamma}}_i) = \frac{\sum_k \sum_t \|J^{\text{offset}}_{k,t} - J^{\text{offset}}_{k,t-1}\|_2^2}{N_j(T + T_f - 1)}, \tag{3.10}$$

and $\mathcal{L}_{\text{heatmap}}$ is the average L_2 distance of joints heatmaps,

$$\mathcal{L}_{\text{heatmap}} = \frac{1}{T} \sum_t \|\mathbf{H}_t - \hat{\mathbf{H}}_t\|_2^2, \tag{3.11}$$

where the joints heatmaps are produced by the transformer encoder as is described in Sec. 3.2.4.

Note that although Eq. (3.5) already captures Eq. (3.9) to some extents, it couples the root and the offsets, thus potentially impeding learning. We empirically observe that adding Eq. (3.9) leads to faster convergence. As the camera motion at each frame is unknown and our predicted root joint is relative to the camera coordinate, $\mathcal{L}_{\text{smooth}}$ factors out the root motion and ensures smooth joints motion.

We apply intermediate pose supervision by computing the losses in Eq. (3.7), except for the heatmap loss, for each layer of the decoder to guide the learning. Additionally, we normalize these losses by the number of target trajectories within a batch to maintain approximate consistency in magnitude across different batches.

3.3 Experiments

We evaluate our approach on three datasets: JTA [52], CMU-Panoptic [88] and Posetrack2018 [4].

Evaluation Metrics. Our method involves three tasks, multi-person pose estimation, tracking and motion forecasting. For *3D pose estimation*, Mean Per Joint Position Error (MPJPE) is used to evaluate the 3D pose accuracy in terms of millimeters, which is defined as

$$\text{MPJPE} = \frac{\sum_k \sum_t \hat{V}_{k,t}^{(i)} \cdot \|J_{k,t}^{(\sigma_i)} - \hat{J}_{k,t}^{(i)}\|_2}{\sum_k \sum_t \hat{V}_{k,t}^{(i)}}, \tag{3.12}$$

where σ_i is defined in Eq. (3.3). $\text{MPJPE}^{\text{rel}}$ refers to the MPJPE calculated after aligning the root joint J_{root}, defined as

$$\text{MPJPE}^{\text{rel}} = \frac{\sum_k \sum_t \hat{V}_{k,t}^{(i)} \cdot \|(J_{k,t}^{(\sigma_i)} - J_{\text{root},t}^{(\sigma_i)}) - (\hat{J}_{k,t}^{(i)} - \hat{J}_{\text{root},t}^{(i)})\|_2}{\sum_k \sum_t \hat{V}_{k,t}^{(i)}}. \tag{3.13}$$

To compare with [14], we report the Percentage of Correct 3D Keypoints (3D-PCK), where a joint is considered as correct if its distance from the corresponding target joint is less than 150 millimeters. Formally, 3D-PCK is defined as

$$\text{3D-PCK} = \frac{\sum_k \sum_t \hat{V}_{k,t}^{(i)} \cdot \mathbb{1}(\|(J_{k,t}^{(\sigma_i)} - \hat{J}_{k,t}^{(i)})\|_2 < 150mm)}{\sum_k \sum_t \hat{V}_{k,t}^{(i)}}, \tag{3.14}$$

where $\mathbb{1}(\cdot)$ represents the indicator function, outputting 1 if the condition within the function is met, and 0 otherwise. To compare with [36], [51] on JTA dataset, we also report F1 $@thr \in \{0.4, 0.8, 1.2\}$ meters, where a joint is considered as true positive if its distance from the corresponding target joint is less than thr. Formally, F1 is defined as

$$\begin{aligned}
\text{F1} &= \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}, \\
\text{Precision} &= \frac{\sum_k \sum_t \mathbb{1}(V_{k,t}^{(\sigma_i)} > 0.8) \cdot \mathbb{1}(\|(J_{k,t}^{(\sigma_i)} - \hat{J}_{k,t}^{(i)})\|_2 < thr)}{\sum_k \sum_t \mathbb{1}(V_{k,t}^{(\sigma_i)} > 0.8)}, \\
\text{Recall} &= \frac{\sum_k \sum_t \hat{V}_{k,t}^{(i)} \cdot \mathbb{1}(\|(J_{k,t}^{(\sigma_i)} - \hat{J}_{k,t}^{(i)})\|_2 < thr)}{\sum_k \sum_t \hat{V}_{k,t}^{(i)}}.
\end{aligned} \tag{3.15}$$

Note that the total number of predicted joint is based the probability of visibility $V_{k,t}^{(\sigma_i)}$, where we use a threshold of 0.8 to determine whether a joint is visible or not, represented as $\mathbb{1}(V_{k,t}^{(\sigma_i)} > 0.8)$. In this context, Precision denotes the percentage of true positive predictions among all the visible predicted joints, while Recall represents the percentage of true positive predictions among all the visible target joints. To compare with [24], [52], [79], [161], [211], [238], we also report the AP defined in [4] on JTA and Posetrack datasets. Specifically, we first define the 2D or 3D keypoint/joint similarity (OKS) as

$$\mathrm{OKS}_{k,t} = \mathbb{1}(\|J_{k,t}^{(\sigma_i)} - \hat{J}_{k,t}^{(i)}\|_2 < 0.5 * s_{\mathrm{head}}^{(i)}), \tag{3.16}$$

where $s_{\mathrm{head}}^{(i)}$ is the size of head for the person i. Then we have the OKS for the predicted joint $J_{k,t}^{(\sigma_i)}$ along with its visibility score $V_{k,t}^{(\sigma_i)}$ as the confidence of prediction. Next, assume there are m joints in total for one video in the evaluation set. We rank all the predicted joints in descending order based on their visibility scores, and get the corresponding ordered OKS as $\{\mathrm{OKS}_j^{\mathrm{rank}}\}_{j=1}^{m}$. Then we have the Average Precision defined as

$$\mathrm{AP} = \frac{1}{m}\sum_{n=1}^{m}\frac{\sum_{j=1}^{n}\mathrm{OKS}_j^{\mathrm{rank}}}{n}. \tag{3.17}$$

For *tracking*, we follow [4], [167] to report MOTA metrics defined either on 2D or 3D keypoint/joint. Specifically, we assume frame t has $m_{k,t}$ visible target joints k with corresponding tracking ID and $n_{k,t}$ visible predicted joints k. According to Eq. (3.16), we have its similarity between the target and matched predicted joint, denoted as $\{\mathrm{OKS}_{k,t}^{(i)}\}_{i=1}^{m_{k,t}}$. Then, for joint k, we can define the False Negative (FN) as the number of missing detected target joints and False Positive (FP) as the number of incorrectly detected joints, which are formulated as

$$\begin{aligned}
\mathrm{FN}_{k,t} &= m_t - \sum_{i}^{m_t}\mathrm{OKS}_{k,t}^{(i)}, \\
\mathrm{FP}_{k,t} &= m_t - \sum_{i}^{m_t}\mathrm{OKS}_{k,t}^{(i)}, \\
\mathrm{MOTA}_k &= 1 - \frac{\sum_t \mathrm{FN}_{k,t} + \mathrm{FP}_{k,t} + \mathrm{IDS}_{k,t}}{\sum_t m_t},
\end{aligned} \tag{3.18}$$

where IDS refers to the number of times a tracking ID switches between consecutive frames $t-1$ and t. In other words, a target joint that has the same

Method		Pose Estimation					Tracking
		AP	F1@0.4m	F1@0.8m	F1@1.2m	3D-PCK	MOTA
OpenPose(2019) [24]	BU	50.1	-	-	-	-	-
THOPA(2019) [52]	BU	59.3	-	-	-	-	59.3
LoCOn(2020) [51]	BU	-	50.8	64.8	70.4	-	-
PandaNet(2020) [14]	SS	-	-	-	-	83.2	-
Cheng et al.(2021) [36]	BU+TD	-	57.2	68.5	72.9	-	-
Ours (t=4+1)	SS	66.5	56.2	67.9	73.1	83.8	-
Ours (t=4+2)	SS	64.5	53.2	65.9	71.2	82.8	-
Ours (T=1)	SS	65.3	59.7	70.7	75.7	83.4	61.4
Ours (T=4)	SS	**70.5**	**60.3**	**71.5**	**76.4**	**85.7**	**63.2**

Table 3.1: Quantitative results of 3D pose tracking on JTA dataset.

tracking ID in frames $t-1$ and t is matched with a predicted joint that has a different tracking ID. Finally, the MOTA is calculated by averaging over all joints. For *motion forecasting*, we report $\text{MPJPE}^{\text{rel}}$ for 3D pose estimation and 3D path error of root joint, following [25], in Tab. 3.2.

Implementation Details. For feature extraction, we employ ResNet50 as the CNN backbone to create multi-scale features from each image. These features are then chronologically stacked into a feature volume denoted by $\mathbf{F}^l$, where $l = 3, 4, 5$ represents the index of the convolution stage of the respective feature map. A 1×1 convolution transforms these multi-scale feature volumes to have a consistent channel size of $C = 384$.

Within the Snipper framework, we utilize 6 transformer encoder and decoder layers. Each layer is equipped with 8 heads in the deformable attention module, which is centered on the frame. All heads are initialized with the same attention weights, and their initial offsets are uniformly distributed in angular directions, ranging from 0 to 360 degrees.

Snipper is trained on 8 V100 GPUs with a batch size of 16. We employ multiple datasets: JTA [52] at 6FPS, CMU-Panoptic [88] at 3FPS, and Posetrack2018 [4] at 7.5 FPS. Additionally, we incorporate the COCO dataset by applying a 2D transformation to create a video snippet. Across all datasets, we adhere to a 14-joint format, as specified in MPII [6].

Ours (T=1) and Ours (T=4) denote the evaluation of model trained on a snippet of 1 and 4 frames. To ensure a fair comparison, we train and test our model exclusively on the corresponding dataset, in line with previous studies.

To achieve *multi-person tracking over the whole video*, for two consecutive snippets (T=4) consisting of frames $\{t, ..., t+3\}$ and $\{t+3, ..., t+6\}$, the association of tracking ID is based on the common frame $t+3$ with the nearest 3D pose matching measured in Euclidean distance. The process is shown in Fig. 3.10. For snippet (T=1), whole-video tracking is achieved by Hungarian matching on poses of two consecutive frames t and $t+1$. For motion forecasting, T_f=2 is used based on T=4 observed frames. Ours (t=4+1) and Ours (t=4+2) denote the evaluation on predicted pose at the 1st and 2nd future frame. The average inference time for a single snippet of 1 and 4 frames on the JTA dataset is 76ms and 266ms, respectively, on a single V100 GPU. The models have 40M and 43M parameters, respectively.

3.3.1 JTA Evaluation

For the JTA dataset, we adjust the input image resolution to 540×960 and downsample the video to 6 FPS. As no prior work has concurrently evaluated all three tasks on this dataset, we present a comprehensive assessment in Tab. 3.1. Here, we compare our method with state-of-the-art approaches for multi-person 3D pose estimation, tracking, and motion forecasting. For pose estimation, our method outperforms the single-stage PanadaNet [14], registering improvements of 0.2% and 2.5% in 3D-PCK for Ours (T=1) and Ours (T=4), respectively. In comparison with [51] and [36], Ours (T=4) demonstrates an F1@0.4m increase of around 10% and 3%, respectively. For tracking accuracy, Ours (T=4) achieves a roughly 4% improvement in MOTA compared to THOPA [52]. As for motion forecasting, our single-stage architecture competes favorably against existing methods like [24], [33], [52]. Specifically, it outperforms them in predicting future motion for the next two frames, evidenced by F1@0.4m and 3D-PCK scores of 66.5 and 83.8, respectively.

Tab. 3.2 compares our framework with HMP [25], the only other work that forecasts 3D poses from RGB images. "No forecasting" means to keep the last observed pose for evaluation without motion prediction. For fair comparison, we retrain HMP [25] to take only 4 frames as input and forecast the next 2 frames. The evaluation is done for the deterministic mode of HMP. We

Method	3D Path Error (mm)		$\text{MPJPE}^{\text{rel}}$ (mm)	
Forecasting Time	166ms	333ms	166ms	333ms
No forecasting	353.5	409.1	123.5	139.1
HMP^1 [25]	90.3	112.6	35.4	39.5
HMP^2 [25]	94.5	121.8	48.5	61.4
HMP (Hourglass [142])	95.2	123.3	46.8	60.6
Ours	92.3	117.7	37.9	43.0

Table 3.2: Quantitative results of motion forecasting on JTA dataset.

use the ground truth history 2D pose in HMP^1 but added Gaussian noise of $\mathcal{N}(0, 3)$ pixels to the ground truth history 2D pose for HMP^2. HMP (Hourglass), means HMP is trained with 2D poses estimated by Hourglass [142]. Our method jointly estimates the poses in the observed frames and forecasts the future motion, which shows comparable accuracy with noise-free HMP but noticeably outperforms it when adding noise to the history pose or using the estimated poses. This highlights the advantages of jointly solving pose estimation, tracking, and forecasting within our framework. The performance gains we observe are primarily attributable to our effective spatiotemporal deformable attention module.

3.3.2 CMU-Panoptic Evaluation

For multi-person pose estimation or tracking, there are mainly 2 data split protocols used in prior works [14], [167]. Protocol 1 follows [167], where 3 views HD cameras $(3, 13, 23)$ and all the haggling videos of version 1.2 are used. The training and testing video split follows [87], [167], and the evaluation metrics follow [167], [197]. Protocol 2 follows [14], [51], where 4 scenarios (Haggling, Mafia, Ultimatum, Pizza) are selected. The testing set is composed of HD videos of camera 16 and 30, and the training set includes videos of the other 28 cameras. The evaluation metrics follow [14], [51]. Since this dataset's motion speed is slower than that of the JTA dataset, we downsample it to 3 FPS to minimize redundancy during training and use a 540×960 image resolution. We also provide test results of 6 FPS with the model trained on 3 FPS, denoted by Ours (T=4, 6 FPS).

The results of protocol 1 are shown in Tab. 3.3. Our method outper-

Method		Backbone	MPJPE	MPJPE^{rel}	MOTA
VoxelPose(2020) [197]	TD	ResNet50	66.9	51.1	-
TesseTrack(2021) [167]	TD	HRNet	**18.9**	-	76.0
VoxelTrack(2022) [251]	TD	DLA-34	66.4	-	-
Ours (t=4+1)	SS	ResNet50	49.0	40.8	-
Ours (t=4+2)	SS	ResNet50	50.7	41.3	-
Ours (T=4, 6 FPS)	SS	ResNet50	45.1	37.3	80.9
Ours (T=1)	SS	ResNet50	48.4	<u>37.5</u>	<u>78.1</u>
Ours (T=4)	SS	ResNet50	<u>44.3</u>	**37.1**	**81.7**

Table 3.3: Quantitative results on CMU-Panoptic dataset in protocol 1.

forms VoxelPose [197] by 22.6mm (+33%) on MPJPE and 14mm (+27%) on relative MPJPE. As for its following work, VoxelTrack [251], we also exceeds 22.1mm on MPJPE. Compared with TesseTrack [167], Ours (T=4) shows higher pose tracking accuracy (81.7 vs 76.0 of MOTA), but lower accuracy on MPJPE, which might be the reason that TesseTrack uses HRNet as the backbone (around 100M parameters) while our method uses ResNet50 (only 43M parameters).

Method		MPJPE					F1	MOTA
		Hag.	Maf.	Ult.	Piz.	Avg.		
MubyNet(2018) [241]	BU	72.4	78.8	66.8	94.3	72.1	-	-
LoCO(2020) [51]	BU	45	95	58	79	69	89.2	-
PandaNet(2020) [14]	SS	40.6	37.6	**31.3**	55.8	42.7	-	-
Benzine et al.(2021) [15]	SS	70.1	66.6	55.6	78.4	68.5	-	-
Jin et al.(2022) [86]	SS	63.7	58.5	52.3	69.1	60.9	-	-
Wang et al.(2022) [217]	SS	53.3	51.2	49.1	61.5	53.8	-	-
Ours (t=4+1)	SS	41.4	38.8	41.6	44.9	40.3	88.7	-
Ours (t=4+2)	SS	43.0	40.9	42.9	47.4	42.4	85.5	-
Ours (T=4, 6 FPS)	SS	37.3	37.1	39.0	42.6	38.2	90.0	93.0
Ours (T=1)	SS	<u>37.6</u>	<u>38.5</u>	39.7	<u>45.0</u>	<u>39.4</u>	<u>89.4</u>	<u>92.9</u>
Ours (T=4)	SS	**36.8**	**36.9**	<u>38.6</u>	**42.5**	**37.9**	**90.1**	**93.4**

Table 3.4: Quantitative results on CMU-Panoptic dataset in protocol 2.

For protocol 2, we show comparisons with six recent works [14], [15], [51], [86], [217], [241] in Tab. 3.3.2. Snipper outperforms in F1 scores and MPJPE across all six sequences, with the exception of the Ultimatum sequence, where PandaNet [14] achieves 7.3mm lower than ours. For tracking, Snipper achieves over 90% in MOTA. For motion prediction, ours (t=T+1) and ours (t=T+2) in both protocols 1 and 2 yields competitive results on MPJPE, only about 3 and 5mm worse than ours (T=4) with observed motion (see Tab. 3.3.2).

Method		Head	Sho	Elb	Wri	Hip	Kne	Ank	Avg
DetTrack(2020) [211]	TD	84.9	87.4	84.8	79.2	77.6	79.7	75.3	**81.5**
PT_CPN++(2018) [238]	TD	82.4	**88.8**	**86.2**	**79.4**	72.0	**80.6**	**76.2**	80.9
TML++(2019) [79]	BU	-	-	-	-	-	-	-	74.6
STAF(2019) [161]	BU	-	-	-	64.7	-	-	62.0	70.4
Ours (T=1)	SS	86.5	85.6	71.5	67.9	78.1	72.0	62.6	74.9
Ours (T=4)	SS	**86.7**	85.9	71.6	68.6	**78.3**	72.5	63.6	75.3

Table 3.5: Quantitative results (AP) of pose estimation on Posetrack2018 val set.

Method		Head	Sho	Elb	Wri	Hip	Kne	Ank	Avg
DetTrack(2020) [211]	TD	74.2	76.4	**71.2**	**64.1**	64.5	**65.8**	**61.9**	**68.7**
PT_CPN++(2018) [238]	TD	68.8	73.5	65.6	61.2	54.9	64.6	56.7	64.0
Rajasegaran et al.(2021) [162]	TD	-	-	-	-	-	-	-	55.8
Rajasegaran et al.(2022) [163]	TD	-	-	-	-	-	-	-	58.9
TML++(2019) [79]	BU	76.0	76.9	66.1	56.4	65.1	61.6	52.4	65.7
STAF(2019) [161]	BU	-	-	-	-	-	-	-	60.9
Ours (T=1)	SS	82.0	82.0	58.8	53.8	72.3	61.1	40.2	64.2
Ours (T=4)	SS	**82.1**	**82.3**	59.0	53.7	**72.7**	61.7	41.7	64.7

Table 3.6: Quantitative results (MOTA) of tracking on Posetrack2018 val set.

3.3.3 Posetrack2018 Evaluation

We use Posetrack2018 [4] to validate that our method is flexible to 2D pose tracking task by simply skipping joint depth prediction. Since the provided annotations of PoseTrack2018 dataset are in 7.5 FPS, we downsample the input video to 7.5 FPS accordingly in both training and testing. We report our results on the validation set following prior works [79], [161], [211]. Tab. 3.5 and 3.6 present the results of Snipper on 2D pose estimation (AP) and pose tracking (MOTA). When comparing with bottom-up approaches, our method exceeds STAF [161] for about 4% in AP and MOTA, and also shows competitive results with TML++ [79]. For the most recent top-down methods [211], our method is around 6% AP and 4% MOTA worse, which could attribute to the fact that our method jointly performs multi-person detection and pose tracking, while [211] relies on a specialized person detector to obtain the multi-person detection. When comparing with the most recent works [162], [163], our method still achieves around 8.9% and 5.8% MOTA better, which is largely credited to the inclusion of pose estimation that helps robust tracking.

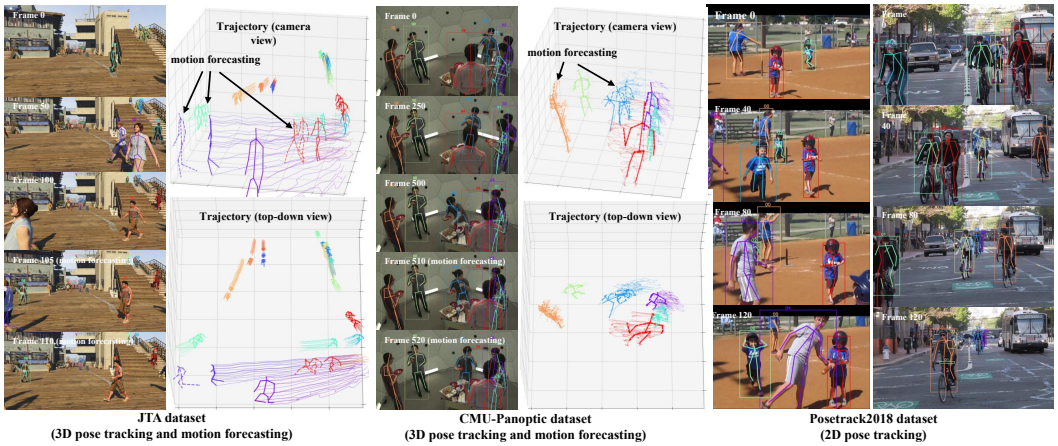


Figure 3.9: Qualitative results on JTA, CMU Panoptic and Posetrack2018 datasets.

3.3.4 Ablation Study

We present several key ablation studies (discussed in Sec. 3.2.3) on JTA dataset to highlight the effectiveness and efficiency of our proposed spatiotemporal deformable attention and the correlation between pose estimation and tracking in Tab. 3.7. The comparison of five attention strategies in terms of training time vs. performance is shown in Fig. 3.11 to illustrate the effectiveness and efficiency of our proposed attention module.

Method	3D Pose Estimation					Tracking
	AP	F1@0.4m	F1@0.8m	F1@1.2m	3D-PCK	MOTA
Self attention	53.2	42.0	55.0	62.4	71.0	49.9
Spatial deform. att.	69.0	53.8	69.3	75.3	84.4	62.3
Direct 3D sampl.	62.8	54.3	65.0	71.6	83.2	55.9
Entire snippet	71.2	59.3	70.3	76.2	85.0	63.4
Ours (single scale)	54.5	43.1	56.8	64.1	73.4	50.2
Ours (2D bbx tracking)	66.5	58.1	69.0	71.8	81.9	54.6
Ours (2D pose tracking)	67.1	59.5	69.3	73.4	83.3	55.9
Ours (w/o smooth loss)	69.1	59.7	70.7	75.9	86.1	62.4
Ours (w/o temp. enc.)	67.3	56.5	68.3	73.7	84.9	55.0
Ours (trajectory query)	69.3	58.6	71.1	76.3	85.0	62.9
Ours (w/o offset norm.)	69.1	57.1	70.8	75.2	78.5	62.4
Ours (encoder layers 4)	67.1	58.0	69.3	74.4	85.3	60.6
Ours (encoder layers 2)	61.8	52.2	63.6	69.1	81.7	53.4
Ours (w/o heatmap)	68.8	57.4	70.1	74.8	82.6	61.6
Ours (1 FPS)	68.5	58.8	69.1	73.4	84.0	62.1
Ours (30 FPS)	69.2	59.2	70.5	75.2	84.2	62.7
Ours	70.5	60.3	71.5	76.4	85.7	63.2

Table 3.7: Quantitative results of ablation study on JTA dataset.

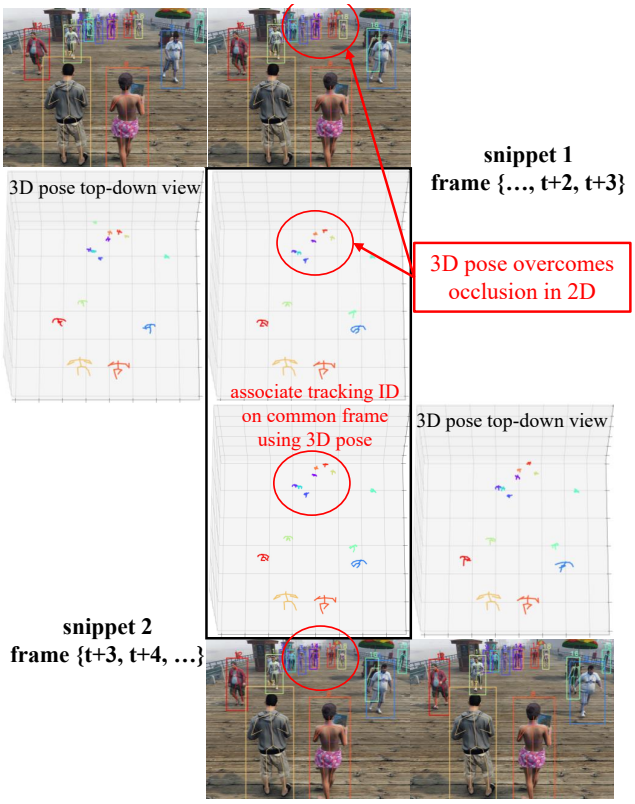


Figure 3.10: Association between two consecutive snippets.

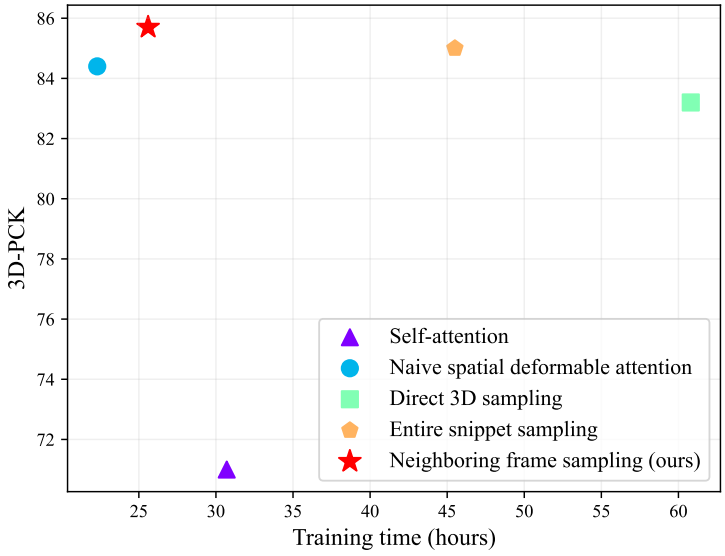


Figure 3.11: Comparison of five attention strategies in terms of training time v.s. performance (3D-PCK).

Self-attention [216]. Our method outperforms self-attention (more than 20% in all metrics), which can be attributed to two main factors: 1) destruction of spatiotemporal local context within the multi-frame features and 2) prohibitive compute for high-resolution multi-scale image features needed to estimate pose of small people.

Naïve spatial deformable attention [265]. This strategy performs slightly worse than ours, especially in accurate pose estimation (53.8 vs 60.3 of F1@0.4m) as it does not account for scene, camera, or object motions over time, where the temporal features at the same image position are not corresponding.

Direct 3D sampling. This strategy also outperforms self-attention, showing that deformable attention in 3D space is an essential technique to encode spatiotemporal information from multi-frame features. However, it is still worse than ours by a large margin, mainly because the interpolation in temporal domain is ill-defined without known correspondences between frames such as optical flow, leading to the aggregated features incorrect in time.

Entire snippet sampling. Our mechanism requires less computation cost and takes only 50% training time of the model with attention over the entire snippet, but gives similar performance, which showcases the efficiency of our approach.

Ours (2D pose tracking) and Ours (bbx tracking) denote that the association between snippets is based on 2D pose or 2D bounding box instead of 3D pose, which reduces 7.3% on MOTA and 2% on 3D-PCK for 2D pose and 8.6% and 3.4% for 2D bounding box. Compared to 2D pose, 3D pose alleviates the issues of occlusion between people on an image since depth information is considered, which demonstrates that effective pose estimation facilitates tracking.

Ours (w/o smooth loss) denotes the model trained without joint smoothness loss $\mathcal{L}_{\text{smooth}}$ in Eq. 3.7 for a snippet of 4 frames as the input. The pose scores are lower than our full model, illustrating that contextual information for tracking also helps improve pose estimation. However, this model without the smoothness loss still performs better than our model but with 1-frame

input, indicating the effectiveness of the spatiotemporal attention model.

Ours (1 FPS) and Ours (30 FPS) denote the model trained with video snippets at 1 FPS and 30 FPS respectively. The pose tracking performances are slightly worse than using 6 FPS, which can be the factor that very high frame rate introduces too much redundancy and easily results in overfitting during training, while very low frame rate introduces too much discrepancy and results in missing clues for tracking and forecasting.

No temporal encoding is proposed to make the pose queries aware of the chronological order within a trajectory. Without temporal encoding, the pose tracking accuracy decreases by over 8% on MOTA, which may attribute to more frequent person ID switches without trajectory temporal encoding.

Trajectory query. In Snipper, there are $N(T + T_f)$ queries with each query focusing on *the pose of each queried person at each time*. To illustrate the effectiveness of $N(T+T_f)$ queries strategy, we compare it with the strategy of N queries where each query focuses on *the trajectory of each queried person*. We can see from Tab. 3.7 that N queries strategy produces worse accuracy of pose tracking (69.3 vs. 70.5 for AP and 62.9 vs. 63.2 for MOTA) since there is bottle-neck between the dimension of query embedding (each embedding is of size 384) and pose trajectory (each $T = 4$ trajectory is of size 240 with 15 3D joints and visibility), especially for large T.

W/o offsets normalization. For 3D pose estimation, we propose to normalize 2D joint offsets by the depth to overcome the issue of scale on 2D image. The accuracy reduction, especially the 3D-PCK (-7.2%), illustrates the effectiveness of our proposed normalization strategy.

Number of layers of encoder. The transformer encoder is the key component to encode spatiotemporal features of the input snippet. The number of layers in the encoder is able to illustrate the effectiveness of spatiotemporal deformable attention to encode these features, as is shown in Tab. 3.7.

Heatmap supervision. This corresponds to supervising the first N_J channels of each temporal slice in the volume to be the multi-person joints' heatmap. As can be seen in Tab. 3.7, this intermediate supervision improves the 3D pose accuracy (2.9% of 3D-PCK).

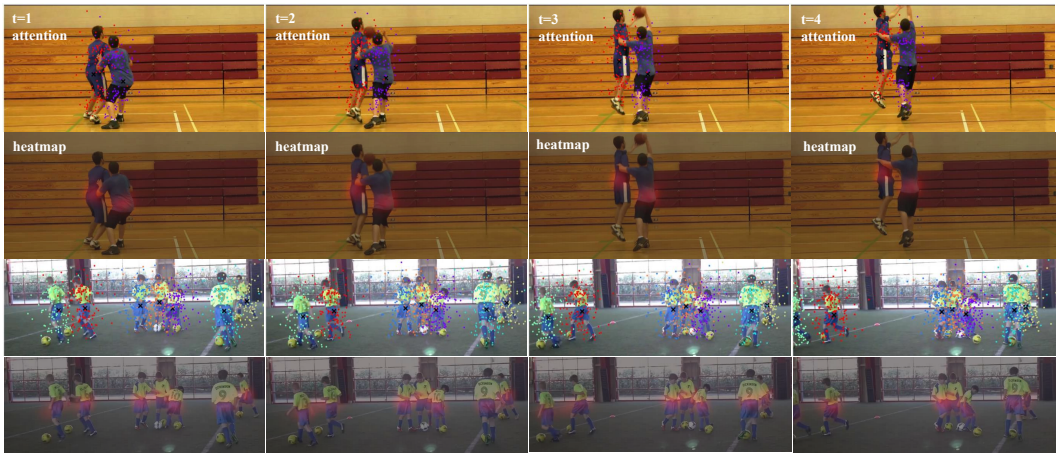


Figure 3.12: Visualization of deformable attention in the transformer decoder and heatmaps of root joint.

3.3.5 Discussion

Correlations among three tasks. Pose estimation and tracking are correlated with each other. Accurate 3D poses facilitate tracking, while robust tracking provides informative temporal clues for pose estimation within the snippet. This is validated by consistently better pose estimation and tracking performance of Ours (T=4) than Ours (T=1) or multi-stage methods [51], [163], [197], [206] on the three datasets in Tab. 3.1, 3.3, 3.3.2, 3.5 and 3.6.

On the other hand, pose tracking builds the crucial history for motion forecasting demonstrated by the better forecast motion of *Ours* than no forecasting in Tab. 3.2. Though in this paper, we cannot demonstrate that motion forecasting in turn helps pose tracking, as the performance of pose tracking does not increase when we include an extra task of future motion forecasting in our framework. We include the important task of motion forecasting for two additional purposes: efficiency and robustness. For efficiency, the encoded spatiotemporal features within the video snippet capture crucial history for motion forecasting. Thus, we can address the three tasks in a single stage with our unified framework for efficiency, i.e., *running one network vs. many networks*. TesseTrack [167] (∼100M params) addresses only pose tracking, while our method (only 43M params) addresses all three tasks. For robustness, existing work, such as HMP [25], on motion forecasting uses off-the-shelf

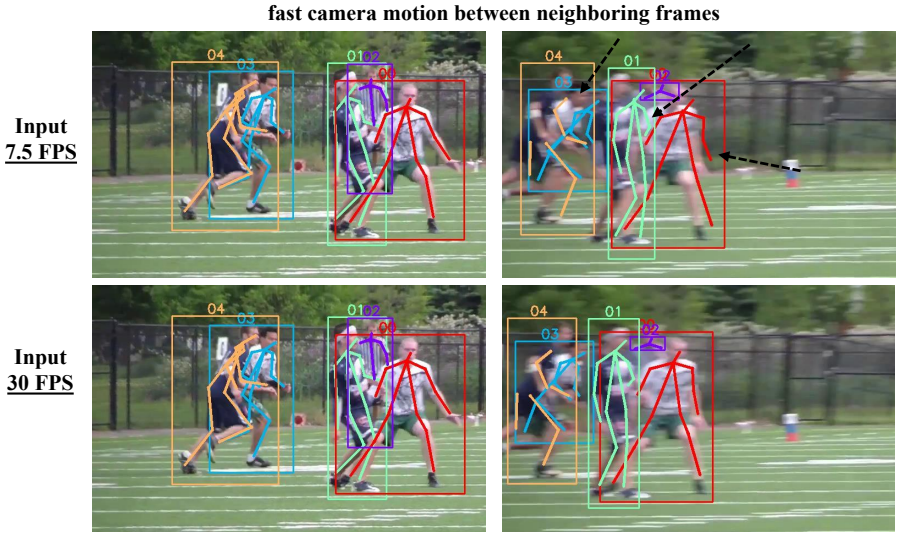


Figure 3.13: Failure cases.

pose estimators for the history pose estimation. But the pose estimators could fail unpredictably and thus cause forecasting to fail. Solving them jointly can be robust to pose tracking errors. We validate that our unified framework helps motion forecasting in Tab. 3.2, where our method shows better performance than the multi-stage method, HMP (Hourglass [142]).

Generalization ability. To validate the generalization ability of our method, we train our model on a hybrid of datasets, MuCo [128], COCO [112], Posetrack2018 [4] and JTA [52]. Then we test our model to predict 3D pose tracking on the unseen MuPoTS [128] and Posetrack val set (no 3D pose annotations). The predicted 3D pose is smooth and the tracking is consistent across the entire videos, even in occlusion cases. The qualitative results are included in the supplementary video [2].

Attention visualization. We visualize the attention maps of the last layer in the transformer decoder in Fig. 3.12, where the sampling positions for each person's pose queries are presented by the same color and "×" means the root joint position. Our proposed spatiotemporal deformable attention samples around each person's whole body and later aggregates these sampled features together to update pose queries. Compared with self-attention, our

[2]Link of supplementary video.

proposed deformable attention typically preserves better spatial and temporal relationships for correct pose regression, and compared with the naïve spatial deformable attention, our method considers the motion changes between frames as is indicated by the root joint positions "×" of each person in the video snippet in Fig. 3.12.

Failure case and limitations. Our method could suffer from fast camera movements as shown in Fig. 3.13, where there are quite large discrepancies between neighboring frames at 7.5 FPS. In this special case, the problem can be alleviated when we use video snippet at 30 FPS as the input due to preserve the correlation between frames. Another limitation lies in the spatiotemporal positional encoding that does not consider the correspondence between frames, especially when the camera motion exists between frames. Future work could explore the approach to explicitly use the camera poses such as ray-based position encoding and the low-resolution optical flow to guide the feature aggregation.

3.4 Conclusion

We present Snipper in the chapter, a unified spatiotemporal transformer for simultaneous multi-person 3D pose estimation, tracking and motion forecasting on a video snippet. We propose an efficient yet powerful spatiotemporal deformable attention module to aggregate spatiotemporal information across the snippet. We demonstrate the effectiveness of Snipper on three challenging public datasets where a generic Snipper model rivals specialized state-of-the-art baselines trained for 3D pose estimation, tracking, and forecasting.

Chapter 4

Event-based 3D Human Pose and Shape Estimation

4.1 Introduction

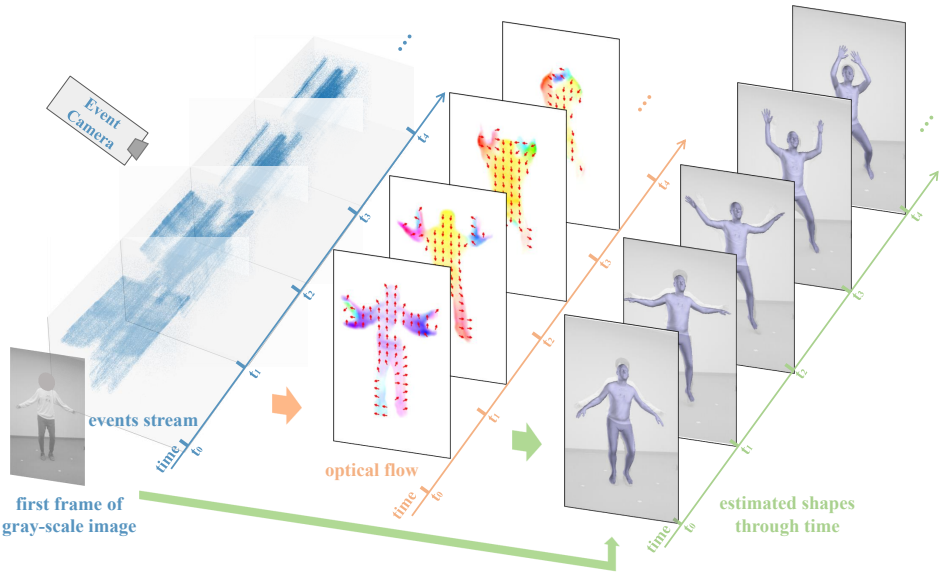


Figure 4.1: An overview of our approach, EventHPE.

Human pose and shape estimation plays a pivotal role in the field of computer vision, drawing significant research interest over the years [136], [176]. Traditional approaches primarily rely on conventional RGB cameras for image capture [91], [93], [98], [228], [264]. However, the advent of event cameras introduces a transformative imaging paradigm [59]. Unlike frame-based cameras, event cameras asynchronously measure changes in pixel brightness,

making them particularly adept at capturing localized object motion. This innovation has sparked research across various computer vision applications, including camera pose estimation [60], gesture recognition [3], and 3D reconstruction [166]. Moreover, it has attracted commercial interest in diverse application areas such as robotics, augmented and virtual reality, and autonomous driving [59]. Despite these advancements, the potential of event cameras in 3D human pose and shape estimation remains largely untapped.

DHP19 represents an early attempt to estimate 2D human poses, treating a cluster of events as a dense, frame-based image [23]. A more recent initiative, EventCap, takes a step further by capturing 3D human motions using an event camera [227]. However, EventCap depends not only on event data but also on an auxiliary sequence of gray-scale images for initial pose estimation at each time step. This limitation prompts us to explore the feasibility of using events as the primary input source for estimating 3D human poses over time, assuming that the beginning pose and shape are either known or can be extracted from the first gray-scale frame. Fig. 4.1 outlines our two-stage methodology, named EventHPE.

Given that both event data and optical flow are intrinsically linked to human motion, and that optical flow offers explicit geometric information to characterize body movements, our framework incorporates optical flow inference from events, referred to as FlowNet, in its initial stage. This stage is trained without supervision. The integration of optical flow allows us to primarily rely on event data for estimating human poses and shapes, thereby eliminating the need for a supplementary stream of gray-scale images. In the subsequent stage of our framework, called ShapeNet, the focus shifts to estimating temporal parametric shape variations using both the event data and the inferred optical flows as inputs. We introduce a novel flow coherence loss to ensure consistency between two modalities: the event-based flow represented by optical flow, and the shape-based flow represented by the movement of vertices on the human body. Both types of flow emerge from the same underlying human motion, making coherence between them crucial.

Our main contributions are summarized below.

- We introduce a novel approach to the challenging task of estimating 3D human parametric pose and shape primarily using event data. Our approach utilizes optical flow, inferred from events, to minimize the dependency on additional gray-scale image sequences. We also implement a unique coherence loss to enforce consistency between two types of motion representation: event-based flow, indicated by optical flow, and shape-based flow, marked by the movement of vertices on the human body. Empirical assessments reveal that our method outperforms several existing state-of-the-art methods.

- We introduce a home-grown dataset called the Multi-Modality Human Pose and Shape Dataset (MMHPSD) [1]. Comprising 240k frames, each frame in MMHPSD contains 12 images captured through various imaging modalities, including an event camera. To the best of our knowledge, MMHPSD is the largest event-based 3D human pose and shape dataset currently available. It is also the first dataset of its kind to be publicly released, as the EventCap dataset [227] is not publicly accessible. The multi-modal nature of MMHPSD enhances its utility, making it a valuable resource for both existing and emerging lines of research.

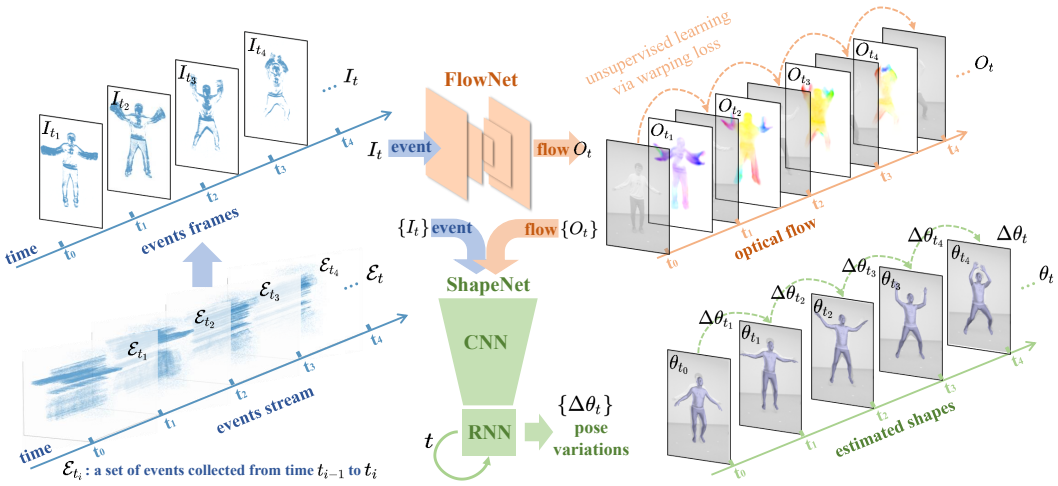


Figure 4.2: Pipeline of our EventHPE framework that consists of two stages, FlowNet and ShapeNet.

[1] Our code and dataset are available at https://github.com/JimmyZou/EventHPE.

4.2 Our EventHPE Approach

Event cameras generate a continuous stream of event signals, where each event is a triplet denoted as $(\mathbf{x}, t, p)$. These cameras also typically output low-frame-rate gray-scale images. In our work, this event stream is partitioned into a sequence of N event packets, expressed as $\mathcal{E} = \{\mathcal{E}_{t_i}\}_{i=1}^{N}$. Each packet, $\mathcal{E}_{t_i}$, contains the events collected within the time interval from t_{i-1} to t_i, as illustrated in Fig. 4.2. Subsequently, each event packet $\mathcal{E}_{t_i}$ is divided into M temporally-ordered subsets. These subsets are aggregated to form individual channels of an event frame, I_{t_i} [62]. As a result, each event frame I_{t_i} is composed of M channels, arranged in temporal sequence. Our approach to structuring these temporal channels is both straightforward and effective, enabling the inclusion of crucial temporal information for human pose estimation. Additionally, we assume the event camera to be static, with known intrinsic parameters.

Parametric human pose and shape is represented by SMPL model [116]. Represented as a differentiable function $\mathbf{v} = \mathcal{M}(\boldsymbol{\beta}, \boldsymbol{\theta}) \in \mathbb{R}^{6,890\times 3}$, the SMPL model takes shape and pose parameters, $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ respectively, and outputs a triangular mesh $\mathbf{v}$ composed of 6,890 vertices. The shape parameters, $\boldsymbol{\beta} \in \mathbb{R}^{10}$, act as the linear coefficients of a Principal Component Analysis (PCA) shape space, which primarily determines individual body characteristics like height, weight, and body proportions. This PCA space is learned from a large dataset of minimally-clothed body scans. On the other hand, the pose parameters $\boldsymbol{\theta} \in \mathbb{R}^{72}$ mainly describe the articulated pose. They consist of a global body rotation and the relative rotations of 24 joints, represented in axis-angle format. The parametric human shape is generated by first applying both shape-dependent and pose-dependent deformations to a template body. Subsequently, forward kinematics articulates this template into its target pose. The surface mesh is then deformed via linear blend skinning. Concurrently, both 3D and 2D joint positions, denoted as $\mathbf{J}_{3\mathrm{D}}$ and $\mathbf{J}_{2\mathrm{D}}$, can be derived through linear regression from the output mesh vertices.

Our method, EventHPE, is summarized in Fig. 4.2, which consists of two

stages. 1) The initial stage focuses on optical flow inference from input events and is detailed in Sec. 4.2.1. Here, the event stream is transformed into a sequence of event frames, which are then input into a CNN model for optical flow prediction. 2) The subsequent stage deals with temporal parametric shape estimation and is elaborated in Sec. 4.2.2. In this phase, a series of event frames, along with their corresponding optical flows, are processed through another CNN model. This extracts vectorized feature representations, which are subsequently input into a Recurrent Neural Network (RNN) for estimating both pose and global translation variations over time.

In our framework, we assume that the beginning pose and shape are known. If they are not, they can be estimated using pre-trained CNN-based methods, such as VIBE [98], akin to previous approaches like EventCap [227]. Importantly, we require the pose and shape information solely for the starting time point or the first gray-scale frame generated by event cameras. Following this, the time-sequenced parametric shapes can be derived accordingly with the estimated pose and translation variations.

4.2.1 Unsupervised Learning of Optical Flow

Building on the observation that both events and optical flow are intrinsically linked to human motion on an image, we propose to incorporate the optical flow inferred from events to furnish additional geometric insights. Specifically, we employ a CNN model, referred to as FlowNet, to predict the optical flow O_{t_i}, using the event frame I_{t_i} as input.

The architecture of FlowNet resembles an encoder-decoder structure and is trained via unsupervised learning. The training employs a warping loss calculated between two consecutive gray-scale images $(I_{t_{i-1}}, I_{t_i})$. Similar to existing work [263], the model uses two types of loss functions: photometric loss and smoothness loss. Upon obtaining the predicted optical flow O_{t_i}, the warped image $\hat{I}_{t_{i-1}}$ can be generated by transforming the second image I_{t_i} to align with the first image $I_{t_{i-1}}$ through bilinear sampling. The photometric loss quantifies the disparity between $I_{t_{i-1}}$ and the warped image $\hat{I}_{t_{i-1}}$. We

define the warping process and the corresponding loss as

$$\mathcal{L}_{\text{photo}}(u, v; I_{t_{i-1}}, I_{t_i}, O_{t_i}) = \sum_{x,y} \rho(I_{t_{i-1}}(x, y) - I_{t_i}(x + u(x, y), y + v(x, y))), \tag{4.1}$$

where ρ is the Charbonnier loss function defined as $\rho(x) = \sqrt{x^2 + \epsilon^2}$ with ϵ being a small constant to make sure the loss function is always non-zero, and (u, v) is the 2D direction of the predicted flow O_{t_i} at pixel (x, y). The Charbonnier loss has been demonstrated to offer greater robustness compared to a simple absolute difference [263]. On the other hand, the smoothness loss serves to regularize the output flow. It achieves this by minimizing the disparity between the flow at each individual pixel and the flows at its neighboring pixels,

$$\mathcal{L}_{\text{smooth}}(u, v; O_{t_i}) = \sum_{x,y} \sum_{i,j \in \mathcal{N}(x,y)} \rho(u(x, y) - u(i, j)) + \rho(v(x, y) - v(i, j)), \tag{4.2}$$

where $\mathcal{N}(x, y)$ is the neighbors of pixel (x, y).

To summarize, FlowNet is trained by minimizing the loss

$$\mathcal{L}_{\text{optical-flow}} = \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{photo}}. \tag{4.3}$$

4.2.2 Pose and Shape Estimation

For each time interval (t_{i-1}, t_i), the event frame I_{t_i} and its associated optical flow O_{t_i} are concatenated. This combined data is then processed through a CNN model to generate a vectorized feature representation. Subsequently, a sequence of these time-sensitive features is fed into a Gated Recurrent Unit (GRU) model, which yields the target outputs. These outputs include inter-frame pose variations, represented as $\Delta\hat{\boldsymbol{\theta}}_{t_i} \in \mathbb{R}^{144}$, and global translation variations, denoted by $\Delta\hat{\mathbf{d}}_{t_i} \in \mathbb{R}^3$, for each time interval. Once the beginning pose and shape are provided, the estimated parametric shapes can be sequentially derived.

Specifically, the estimated global translation $\hat{\mathbf{d}}_{t_i}$ at time t_i can be calculated using the formula $\hat{\mathbf{d}}_{t_i} = \hat{\mathbf{d}}_{t_{i-1}} + \Delta\hat{\mathbf{d}}_{t_i}$. This equation gives rise to the global

translation loss as

$$\mathcal{L}_{\text{trans}} = \sum_{t_i} \|\mathbf{d}_{t_i} - \hat{\mathbf{d}}_{t_i}\|_2^2, \tag{4.4}$$

where $\mathbf{d}_{t_i}$ is the target global translation at time t_i.

As for the estimated pose $\hat{\boldsymbol{\theta}}_{t_i}$ at time t_i, we propose to adopt a 6D rotation representation instead of the traditional 3D axis-angle representation used in the SMPL model. This alternative has been shown to offer superior performance in human pose estimation [261]. Consequently, the output for inter-frame pose variations, $\Delta\hat{\boldsymbol{\theta}}$, is a 144-dimensional vector, where each of the 24 joints is represented by a 6-dimensional vector. Given the estimated pose variations $\Delta\hat{\boldsymbol{\theta}}_{t_i}$, the j-th relative rotation at time t_i can be expressed as follows:

$$\hat{\boldsymbol{\theta}}^j_{t_i} = \mathbf{R}^{-1}\big(\mathbf{R}(\Delta\hat{\boldsymbol{\theta}}^j_{t_i})\mathbf{R}(\hat{\boldsymbol{\theta}}^j_{t_{i-1}})\big), \tag{4.5}$$

where $\mathbf{R}(\cdot)$ is the function converting the 6D rotational representation into a 3×3 rotation matrix. Instead of employing the Euclidean distance metric, we propose the use of the geodesic distance within the $SO(3)$ group to evaluate the discrepancy between the predicted and target poses, which is defined as follows:

$$\mathcal{L}_{\text{pose}} = \sum_{t_i} \sum_{j} \arccos^2\Big(\frac{\text{Tr}\left(\mathbf{R}(\boldsymbol{\theta}^j_{t_i})^\top \mathbf{R}(\hat{\boldsymbol{\theta}}^j_{t_i})\right) - 1}{2}\Big). \tag{4.6}$$

We additionally take into account the positional errors of both the 3D and 2D joints, which are quantified as follows:

$$\mathcal{L}_{\text{3D}} = \sum_{t_i} \sum_{j} \|\mathbf{J}^j_{\text{3D},t_i} - \hat{\mathbf{J}}^j_{\text{3D},t_i}\|_2^2, \tag{4.7}$$

$$\mathcal{L}_{\text{2D}} = \sum_{t_i} \sum_{j} \|\mathbf{J}^j_{\text{2D},t_i} - \pi(\hat{\mathbf{J}}^j_{\text{3D},t_i})\|_2^2, \tag{4.8}$$

where $\hat{\mathbf{J}}^j_{\text{3D},t_i}$ is the predicted 3D position of joint j at time t_i and $\pi(\cdot)$ serves as the projection function, utilizing default camera intrinsic parameters. This measure helps to minimize the accuracy of the joint position estimations in both 3D and 2D spaces.

Finally, we propose a novel coherence loss between two types of flows, *event-based flow* (optical flow) and *shape-based flow* (movement of vertices on the

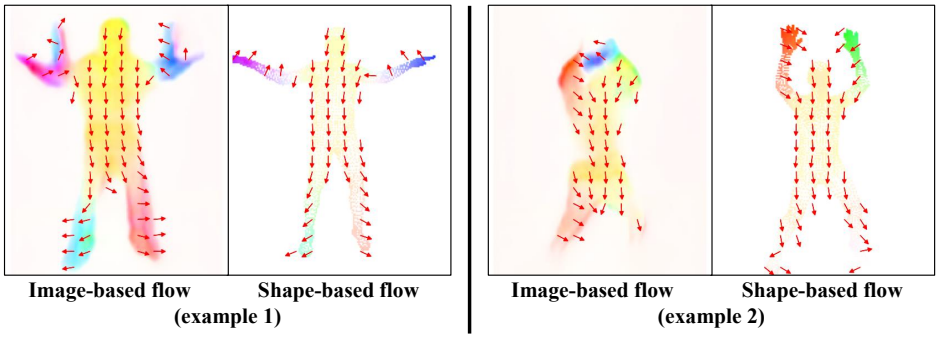


Figure 4.3: An illustration of event-based and shape-based flows.

human body shape). Both types of flow originate from the same underlying human motion, making it crucial to ensure their consistency for more accurate motion estimation. The optical flow O_{t_i}, as detailed in Sec. 4.2.1, represents the event-based flow. On the other hand, the shape-based flow is derived by projecting sequential human body shapes onto the image plane and then calculating the vertex 2D displacements, which is defined as

$$\mathbf{F}_{t_i}^{\text{shape}} = \pi(\hat{\mathbf{v}}_{t_i}) - \pi(\hat{\mathbf{v}}_{t_{i-1}}), \tag{4.9}$$

where $\mathbf{F}_{t_i}^{\text{shape}} \in \mathbb{R}^{6890\times 2}$. In parallel, the event-based flow corresponding to the shape vertices is acquired through bilinear sampling on the optical flow, defined as

$$\mathbf{F}_{t_i}^{\text{img}} = \text{BilinearSample}(O_{t_i}, \pi(\hat{\mathbf{v}}_{t_{i-1}})). \tag{4.10}$$

The coherence loss is then quantified as the cosine distance between the two types of flows:

$$\mathcal{L}_{\text{flow}} = \sum_{t_i}\sum_{v} \frac{\langle \mathbf{F}_{t_i,v}^{\text{shape}}, \mathbf{F}_{t_i,v}^{\text{img}} \rangle}{\|\mathbf{F}_{t_i,v}^{\text{shape}}\|_2 \cdot \|\mathbf{F}_{t_i,v}^{\text{img}}\|_2}, \tag{4.11}$$

where v is the index of vertices and $\langle\cdot\rangle$ means the inner product.

In summary, we train the ShapeNet by minimizing the loss

$$\mathcal{L} = \lambda_{\text{trans}}\mathcal{L}_{\text{trans}} + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{3D}}\mathcal{L}_{\text{3D}} + \lambda_{\text{2D}}\mathcal{L}_{\text{2D}} + \lambda_{\text{flow}}\mathcal{L}_{\text{flow}}, \tag{4.12}$$

where $\lambda_{\text{pose}}, \lambda_{\text{trans}}, \lambda_{\text{3D}}, \lambda_{\text{2D}}$ and λ_{flow} are the weights of corresponding losses.

4.2.3 Our MMHPSD Dataset

We have curated an in-house multi-modality dataset, MMHPSD, specifically for the empirical evaluation of our approach. This dataset fills a significant gap in available resources, as the only existing comparable dataset, EventCap [227], is not publicly accessible.

Data Acquisition. During the data acquisition phase, we employ a multi-camera system containing 4 distinct imaging modalities: an event camera, a polarization camera, and five RGB-Depth cameras. The layout of the camera system is illustrated in Fig. 4.4. Specifically, the event camera is CeleX-V with resolution 1280×800 [30]. *Frame-based camera images are soft-synchronized with the gray-scale images generated by the event camera, and events between consecutive gray-scale images are collected synchronously.* We engage 15 participants for data collection, comprised of 11 males and 4 females. Each subject performs 3 groups of actions (21 different actions in total) for 4 times, where each group includes actions of fast/medium/slow speed respectively. Consequently, we acquired 12 short video clips from each participant. Each clip consists of approximately 1,300 frames and lasts about 1.5 minutes, captured at a frame rate of 15 FPS. The dataset exhibits an average event rate of approximately one million events per second. In total, MMHPSD contains 240k frames, each featuring a gray-scale image, inter-frame events, a polarization image, and five-view RGB and depth images.

Annotation. Annotations for SMPL shape and pose primarily rely on data from the five-view RGB-Depth cameras. For each frame, OpenPose [24] is used to detect 2D joints within the RGB images from all five views. The depth information for these joints is subsequently acquired by mapping the corresponding depth images onto the RGB images. We then average the initial 3D poses across the five views and fit these skeletal poses to the SMPL male model using 3D SMPLify-x [154] to generate initial SMPL parameters. For higher accuracy, the initial shape is further refined. This is achieved by fitting it to a point-cloud assembled from the depth images across the five views, using the L-BFGS optimization algorithm [20]. In this refinement process, the

objective is to iteratively minimize the average distance between each vertex of the SMPL shape and its closest point in the point-cloud. Representative samples and corresponding annotations are depicted in Fig. 4.4.

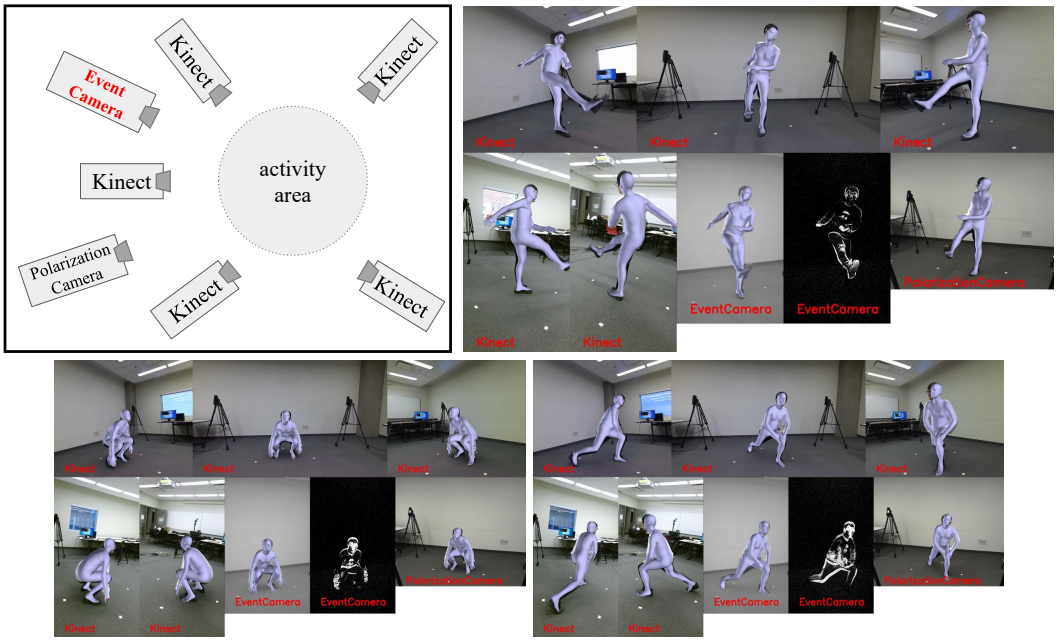


Figure 4.4: Layout of multi-camera acquisition system and examples of our pose and shape annotations.

Dataset Comparison. We provide a comparative analysis of our dataset against two existing event-based human motion datasets in Tab. 4.1. Metrics for comparison include the number of action sequences per subject (Seq#), total number of subjects (Sub#), number of frames (Frame#), availability of annotated poses (Pose) and shapes (Shape), as well as multi-modality (MM). Our dataset excels in terms of the total number of frames and events. While it may not have as many subjects or sequences as the DHP19 dataset, our dataset is unique in its multi-modality nature and the inclusion of precise annotations for both 3D pose and SMPL shape, offering considerable advantages for a variety of research applications.

Dataset	Seq#/Sub#	Frame#	Pose	Shape	MM
DHP19 [23]	33/17	87k	Yes	No	No
EventCap [227]	2/6	-	No	No	No
MMHPSD (ours)	12/15	240k	Yes	Yes	Yes

Table 4.1: A tally of existing event-based human motion datasets.

4.3 Experiments

In this section, we outline the implementation details relevant to training and explain the reported evaluation metrics. Subsequently, we compare our method with various existing baselines, including event-based, frame-based, and video-based methods. We conclude this section with ablation studies to demonstrate the contributions of individual components within our method.

Implementation Details. For training, event packets are converted into 4-channel event frames ($M = 4$), each channel aggregating events over approximately 15 milliseconds. We also tried 1, 2, 8 and empirically found 4 gave better results. These event frames are resized to dimensions of 256×256. During testing, however, we remove the constraint on the temporal length of event packets, allowing it to be dynamically determined by the rate of event generation, as faster motions tend to produce events more quickly. We employ ResNet50 [73] as our backbone CNN architecture and use a single-layer GRU with a hidden dimension of 2048 for ShapeNet's RNN component. The weights $\lambda_{\text{pose}}, \lambda_{\text{trans}}, \lambda_{\text{3D}}, \lambda_{\text{2D}}, \lambda_{\text{flow}}$ are set to $20, 10, 1, 10, 0.1$ respectively. The sequence length for both training and testing in ShapeNet is 16, roughly corresponding to a one-second duration. The batch size for training is set at 16. The learning rate for FlowNet starts at 0.0001 and remains constant for 15 epochs, while that for ShapeNet starts at 0.001 for 5 epochs before decaying to 0.0001 after the third epoch. All models are trained on a single RTX 2080Ti GPU.

Evaluation. Similar to prior studies [91], [98], we evaluate our approach using five metrics: Mean Per Joint Position Error (MPJPE), Procrustes-aligned MPJPE (PA-MPJPE), Pelvis-aligned MPJPE (PEL-MPJPE), Percentage of Correct Keypoints (PCKh@0.5), and Per Vertex Error (PVE).

MPJPE is defined as follows:

$$\text{MPJPE} = \frac{1}{24}\sum_{j=1}^{24} \|\mathbf{J}_{3\text{D}}^{j} - \hat{\mathbf{J}}_{3\text{D}}^{j}\|_2, \tag{4.13}$$

where $\hat{\mathbf{J}}_{3\text{D}}, \mathbf{J}_{3\text{D}} \in \mathbb{R}^{24\times 3}$ are the predicted and target 3D joints. PEL-MPJPE refers to the MPJPE computed after aligning the root joint of both the predicted and the target poses, which is defined as

$$\text{PEL-MPJPE} = \frac{1}{24}\sum_{j=1}^{24} \|(\mathbf{J}_{3\text{D}}^{j} - \mathbf{J}_{\text{root,3D}}) - (\hat{\mathbf{J}}_{3\text{D}}^{j} - \hat{\mathbf{J}}_{\text{root,3D}})\|_2, \tag{4.14}$$

where $\mathbf{J}_{\text{root,3D}}$ and $\hat{\mathbf{J}}_{\text{root,3D}}$ are the root joint positions. PA-MPJPE means MPJPE after performing a rigid transformation, denoted as (R, t), to align the predicted pose with the target pose, which is defined as

$$\text{PA-MPJPE} = \frac{1}{24}\sum_{j=1}^{24} \|\mathbf{J}_{3\text{D}}^{j} - (\hat{\mathbf{J}}_{3\text{D}}^{j} \cdot R + t)\|_2. \tag{4.15}$$

Note that the rigid transformation parameters (R, t) are determined by minimizing the average distance between the target 3D joints $\mathbf{J}_{3\text{D}}$ and the transformed predicted 3D joints $(\hat{\mathbf{J}}_{3\text{D}} \cdot R + t)$. For PCKh@0.5, it denotes the percentage of correctly predicted joints whose PEL-MPJPE is less than 50% of the head length, defined as

$$\text{PCKh@0.5} = \frac{1}{24}\sum_{j=1}^{24} \mathbb{1}(\|(\mathbf{J}_{3\text{D}}^{j} - \mathbf{J}_{\text{root,3D}}) - (\hat{\mathbf{J}}_{3\text{D}}^{j} - \hat{\mathbf{J}}_{\text{root,3D}})\|_2 < 0.5 * s_{\text{head}}), \tag{4.16}$$

where s_{head} is the height of head and $\mathbb{1}(\cdot)$ is the indicator function. PVE is calculated as the Euclidean distance between each vertex on the predicted SMPL mesh and its corresponding vertex on the target mesh, defined as

$$\text{PVE} = \frac{1}{6890}\sum_{i} \|\mathbf{v}^{i} - \hat{\mathbf{v}}^{i}\|_2, \tag{4.17}$$

where $\hat{\mathbf{v}}, \mathbf{v} \in \mathbb{R}^{6890\times 3}$ are the predicted and target SMPL vertices.

4.3.1 Empirical Results

The DHP19 dataset [23] is limited to multi-view event streams and motion-capture data for joints; it lacks gray-scale images and SMPL shape annotations. So we only use this dataset to demonstrate the effectiveness of optical

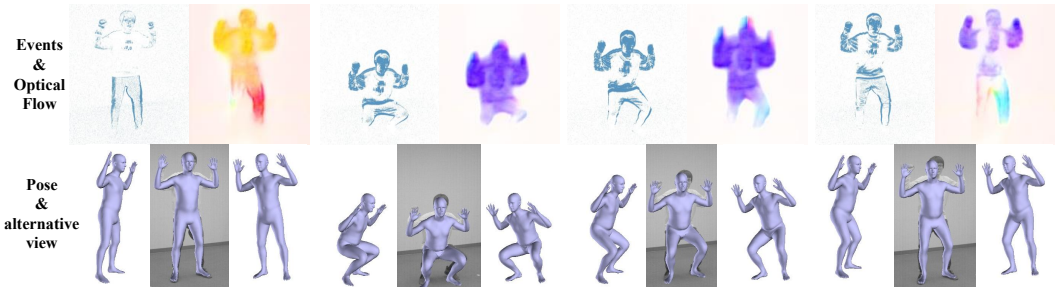


Figure 4.5: A sampled sequence of event frames, corresponding optical flows and the estimated shapes with two alternative views.

Models	Input	MPJPE↓	PA-MPJPE↓	PEL-MPJPE↓	PCKh@0.5↑	PVE↓
DHP19 [23]	E	80.08	74.55	131.73	0.80	-
DHP19 + Flow	E+F	76.76	71.68	130.37	0.82	-

Table 4.2: Quantitative evaluations on DHP19.

flows in event-based pose estimation. We extend the DHP19 method by adding FlowNet-predicted optical flows trained on our MMHPSD dataset, referred to as DHP19+Flow. Quantitative results presented in Tab. 4.2 indicate that the incorporation of optical flows leads to a reduction in joint position errors, as evidenced by a decrease of more than 3mm in both MPJPE and PA-MPJPE metrics. Conversely, PEL-MPJPE does not demonstrate a significant improvement. This discrepancy can be attributed to the fact that our method focuses on detecting 2D joints rather than the entire body shape, meaning that root translation alignment could inadvertently increase distance errors for other joints. Nonetheless, the PCKh metric consistently improves with the inclusion of optical flows as an additional input. These findings validate the effectiveness of our proposal to integrate optical flow for enhanced geometric information extraction from events, thereby improving event-based pose estimation.

MMHPSD dataset offers a rich array of data sources and well-aligned pose and shape annotations, allowing for a comprehensive comparison of our event-based approach with various baseline methods. Quantitative outcomes are detailed in Tab. 4.3, while qualitative insights are visually represented in Fig. 4.6. We also present a supplementary video[2] for better illustration of our results.

[2]Link of supplementary video.

Models	Input	MPJPE↓	PA-MPJPE↓	PEL-MPJPE↓	PCKh@0.5↑	PVE↓
HMR [91]	G	-	64.78	95.32	0.61	-
VIBE [98]	V	-	50.86	73.10	0.76	-
EventCap(HMR) [227]	E+G	-	62.62	89.95	0.64	-
EventCap(VIBE) [227]	E+G	-	50.35	71.85	0.77	-
DHP19 [23]	E	72.42	65.87	74.04	0.81	-
EventHPE(HMR)	E+F	-	53.72	77.80	0.71	-
EventHPE(VIBE)	E+F	-	48.87	69.58	0.79	-
EventHPE	E+F	**71.79**	**43.90**	**54.96**	**0.85**	**53.90**

Table 4.3: Quantitative results on MMHPSD dataset.

HMR [91] serves as a frame-based baseline, while VIBE [98] is employed as a video-based counterpart. It should be noted that both HMR and VIBE rely on a weak camera model that lacks global translation. Additionally, they utilize the neutral SMPL model, in contrast to the male-specific model featured in the MMHPSD dataset. Consequently, we refrain from reporting quantitative evaluations for MPJPE and PVE metrics in these cases.

We consider three additional categories of event-based baselines for comparison. First, DHP19 [23] employs a 2D pose estimation approach using event data, with the ground-truth depths of the detected 2D joints assumed to be known for 3D comparison. Second, EventCap(HMR) and EventCap(VIBE) indicate configurations where HMR or VIBE is integrated into EventCap [227] to infer pose and shape from a sequence of gray-scale images, serving as initial pose estimates over time. Since the original code and evaluation dataset from EventCap are not publicly available, we re-implemented it using the PyTorch L-BFGS optimizer [20], [151] and PyTorch3D differential renderer [165]. Lastly, in our approach, denoted as EventHPE, we have two variants: EventHPE(HMR) and EventHPE(VIBE). These variants use HMR or VIBE to estimate the initial pose and shape based on the first frame of the gray-scale image sequence. These are analogous to the initial pose estimation methods used in EventCap. We also include a version of EventHPE that employs the ground-truth pose and shape as the starting point for evaluation.

For a fair comparison with DHP19, which uses ground-truth depths for detected 2D joints to obtain 3D pose, we contrast its performance against our approach, EventHPE. The quantitative results indicate that EventHPE

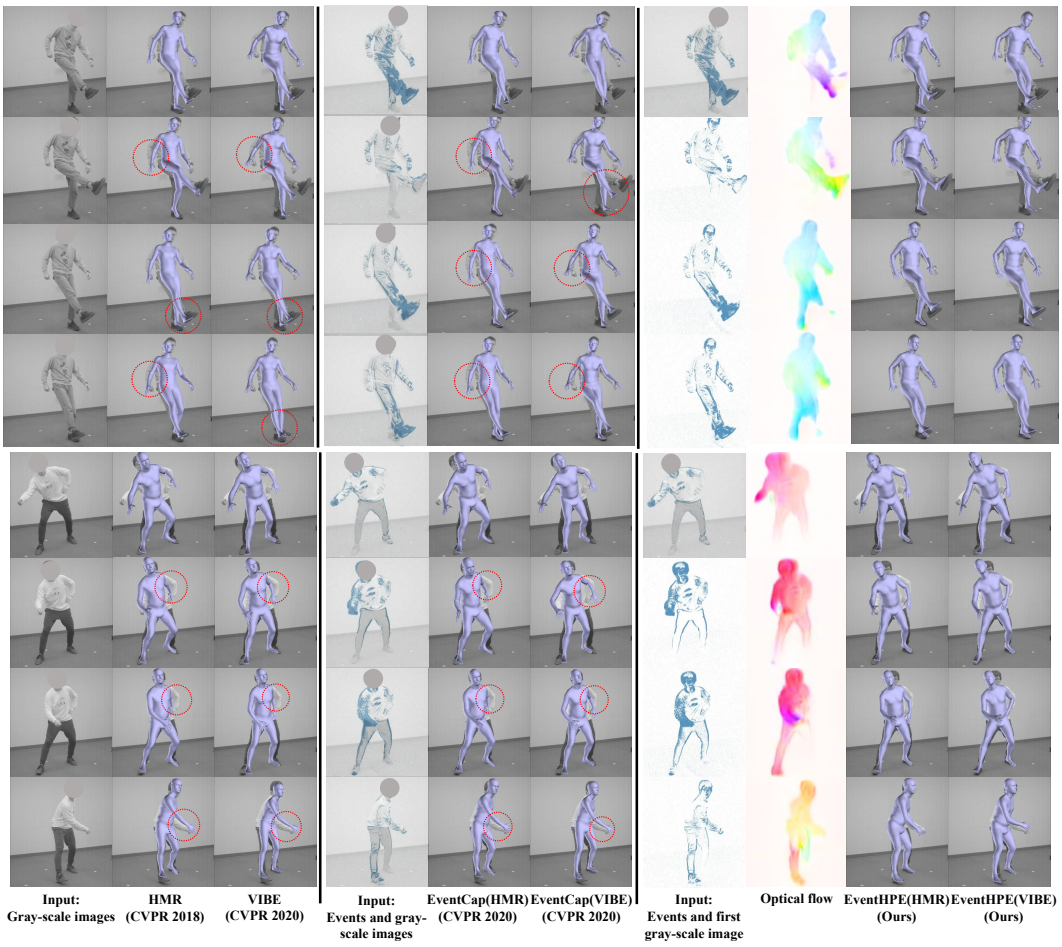


Figure 4.6: Qualitative results on MMHPSD dataset.

outperforms DHP19, showing more than a 20mm reduction in joint errors for PA-MPJPE and PEL-MPJPE metrics, while achieving less than a 1mm error in MPJPE. This improvement can likely be attributed to the design of our method, which predicts the entire body shape subject to topological constraints. In contrast, DHP19 focuses on detecting individual joints in isolation. Consequently, our method yields significantly lower joint errors following alignment procedures.

For a comprehensive comparison between EventCap and our approach, EventHPE, we examine quantitative results across different configurations: EventCap(HMR) vs. EventHPE(HMR), and EventCap(VIBE) vs. EventHPE(VIBE). Notably, our method only requires beginning pose and shape estimation from the first frame of gray-scale images, whereas EventCap ne-

cessitates such estimations across the entire sequence of gray-scale images. Although the accuracy of these initial estimates influences the performance of both methods, EventHPE demonstrates marked improvements in joint errors and PCK metrics.

In the HMR configuration, EventCap(HMR) shows a roughly 2mm reduction in PA-MPJPE, a 5mm reduction in PEL-MPJPE, and a 0.03 increase in PCK compared to HMR alone. In contrast, our method, EventHPE(HMR), achieves improvements of approximately 11mm, 17mm, and 0.1, respectively—over three times the gains observed in EventCap(HMR). A similar trend emerges in the VIBE configuration. EventHPE(VIBE) demonstrates a roughly 2mm improvement in PA-MPJPE, 3.5mm in PEL-MPJPE, and 0.03 in PCK over VIBE alone, while EventCap(VIBE) shows only 0.5mm, 1.5mm, and 0.01 improvements, respectively.

Furthermore, we note that the performance gains in the VIBE configuration are not as pronounced as those in the HMR configuration. This may be due to the already strong performance of VIBE, which leaves less room for improvement. However, both EventCap and EventHPE yield consistently better results in the VIBE setup than in the HMR setup. This suggests that the quality of the initial estimates can significantly impact the overall performance of both methods, but EventHPE is more robust to this variability. This robustness likely stems from our method's reliance solely on events and predicted optical flows as input after establishing initial pose and shape estimates. In contrast, EventCap's performance is constrained by potentially inaccurate initial estimates throughout the sequence of gray-scale images.

The qualitative outcomes presented in Fig. 4.6 further validate the efficacy of our EventHPE method. Eight examples have been selected from two test sequences for illustration. It is evident that even when the beginning pose and shape estimates—obtained from either HMR or VIBE and depicted in the first row of each sequence—are not perfectly aligned or accurate, EventHPE is capable of correcting subsequent predictions to yield well-aligned poses and shapes. In contrast, EventCap demonstrates limited capability for such corrections. This is because EventCap relies on continuous estimates from HMR

Models	Input	MPJPE↓	PA-MPJPE↓	PEL-MPJPE↓	PCKh@0.5↑	PVE↓
EventHPE(w/o flow)	E	80.99	49.43	60.90	0.82	59.77
EventHPE(w/o flow loss)	E+F	78.48	47.36	57.09	0.83	56.58
EventHPE(w/o geodesic)	E+F	77.29	49.02	60.55	0.83	59.84
EventHPE(w/o joints)	E+F	73.79	44.59	55.91	0.84	54.73

Table 4.4: Quantitative results of ablation study on MMHPSD dataset.

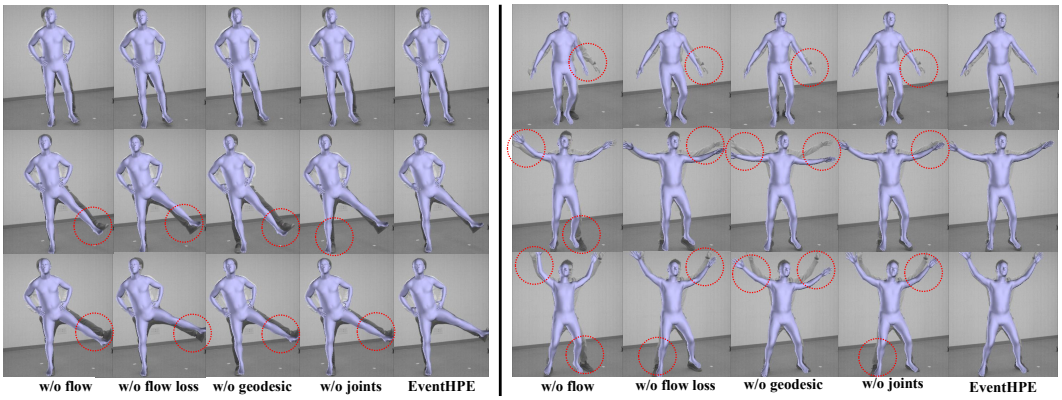


Figure 4.7: Qualitative results of ablation study.

or VIBE for every gray-scale image in the sequence, thus confining its capacity for significant adjustments over time. This qualitative evidence substantiates EventHPE's robustness and ability to refine pose and shape estimates, even when initiated with less-than-ideal conditions.

4.3.2 Ablation Study

In this section, we perform an ablation study to assess the contributions of various components within our EventHPE method. The following variants are considered:

- EventHPE(w/o flow): This variant excludes both the optical flow and the flow coherence loss from our approach.
- EventHPE(w/o flow loss): Here, the optical flow is incorporated into the input, but the flow coherence loss is not applied.
- EventHPE(w/o geodesic): This version employs Euclidean distance in the training process as opposed to the geodesic distance.

- EventHPE(w/o joint): In this configuration, neither 3D nor 2D joint supervision is used during training.

The quantitative and qualitative assessments are presented in Tab. 4.4 and Fig. 4.7. Upon comparing the four ablation variants with the full EventHPE model, several key observations can be made. Specifically, the exclusion of optical flow or geodesic distance leads to an approximate 6mm increase in joint errors as well as shape vertex errors. In the version that omits the flow coherence loss, error rates grow by 3-4mm. On the other hand, removing joint supervision during training results in a more modest increment of 1-2mm in error margins. These results highlight the crucial contribution of each component in enhancing the overall efficacy of the EventHPE model.

The qualitative evaluations presented in Fig. 4.7 further reinforce the significance of incorporating optical flow and geodesic distance into our methodology. The absence of these components compromises the model's ability to align the human body accurately with the background gray-scale images. This underscores the advantage of using geodesic distance over Euclidean distance for quantifying human pose variations in $SO(3)$. Additionally, it indicates that the integration of optical flow, along with flow coherence loss, contributes to a more robust extraction of geometric information essential for accurate human shape estimation from events.

4.4 Conclusion

In this chapter, we have presented a method for estimating parametric human shapes, with a focus on leveraging event-based data. Our empirical results validate the effectiveness and versatility of our approach. However, a limitation of our current framework is its dependency on obtaining an initial pose and shape, either pre-supplied or detected in the first frame of gray-scale images. As future work, we plan to address this constraint by aiming to create an enhanced model that can infer 3D human shapes solely from event-based signals.

Chapter 5

Event-based Human Pose Tracking with SNNs

5.1 Introduction

As we discussed in Chapter 4, visual human pose tracking has attracted increasing research attentions in recent years. While most current research efforts have been focused on RGB cameras [91], [93], [98], [99], [104], [110], [150], [156], [182], [207], [209], [225], [228], [239], [252], [259], event cameras [59], as an emerging vision sensor, present new opportunities in this area. As a novel and biologically-inspired vision system, event cameras are considerably dissimilar to the conventional frame-based cameras. In particular, by adopting its unique asynchronous and independent address-event representation, event cameras are capable of imaging high-speed motions with a very low power consumption. This innovative imaging paradigm has sparked a multitude of research efforts in the field of event-based vision, such as tracking [63], [135], [245], [246], recognition [3], [57], [95], 3D reconstruction [166], [248], and a diverse range of applications in robotics, augmented and virtual reality, and autonomous driving [59].

Recently, data-driven approaches have shown their potential for effective pose estimation from event cameras [23], [170], [227], [266]. One of the earliest approaches [23] uses a CNN model to estimate 2D human poses from event frames. EventCap [227] expands upon this by capturing fast 3D human motion based on a stream of 2D events, as well as a sequence of gray-scale images to

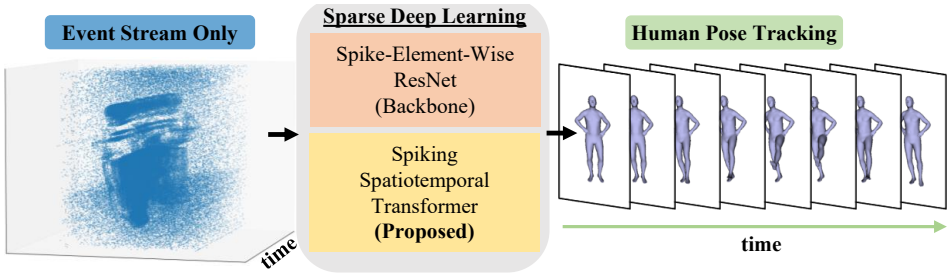


Figure 5.1: Overview of our end-to-end sparse deep learning approach.

establish initial poses over time. Our prior work, EventHPE [266] presented in Chapter 4, reduces the reliance on gray-scale images by using only the first gray-scale frame to extract the starting pose, then relying solely on the event stream for subsequent pose tracking. The concurrent work of [170] instead focuses on static hand pose estimation from event camera, by engaging the ANNs models. Unfortunately, existing methods either require the presence of additional gray-scale images [227], [266], which is not always practical in real-world applications, or treat the event stream as dense frame-based images [23], [170] and input them directly into the ANNs models, which ignores the inherent sparsity of event signals. As a result, the full potential of human pose tracking based only on events remains largely unexploited.

Meanwhile, ANNs, such as ResNet [73] and standard Transformer [200], have demonstrated their competence in various event-based vision tasks [3], [57], [58], [61], [62], [95], [135], [170], [190], [210], [246], [263]. However, compared with dense RGB or gray-scale images, event streams are *spatiotemporally much sparser*, resulting in a growing interest in seeking novel ways dedicated to efficiently process event signals. One promising strategy is based on the SNNs. Unlike traditional ANNs, spiking neurons are employed in SNNs to imitate the event generation process, thus bypassing the unnecessary computations of inactive or non-spiking neurons. Previous efforts have shown the superiority of SNNs in classification tasks, including converting ANNs to SNNs [44], [72], [106], [171], [229], or training SNNs from scratch [57], [58], [107], [235], [236], [262]. There are also attempts [234], [245] proposing a mixed framework of SNNs and ANNs to balance the efficiency and performance in event-based

regression tasks. However, the challenge of conducting pose tracking solely using event signals, and exclusively engaging the SNNs architecture to fully exploit the innate sparsity in events data, remains unaddressed. This may be attributed to the following three challenges. 1) Unlike spike votes used in classification, regression is sensitive to the output values, which may result in additional quantization errors in pose prediction due to the compact spike representation in SNNs. 2) As opposed to high-level label prediction in the static classification tasks, pose tacking requires fine-grained regression of poses over time. 3) Spikes are typically unfolded over time, which naturally preserves only one-directional temporal dependency in SNNs. This may lead to insufficient pose information, especially when the character is not moving in the starting phase and thereby few events are observed for pose estimation.

Motivated by the above observations, our work aims to tackle a relatively new problem of tracking 3D human poses solely based on event streams from an event camera, thus completely eliminating the need for additional input dense images. As presented in Fig. 5.3, our approach is an end-to-end sparse deep learning approach that estimates parametric human poses over time solely from events. This model is entirely built upon SNNs, thus having the promise of being more efficient than that of the dense deep learning models (*i.e.* ANNs). The input event stream goes through a preprocessing step to form a sequence of event voxel grids; SEW-ResNet [57] is then employed as the SNNs backbone to extract pose spike features; this is followed by the proposed Spiking Spatiotemporal Transformer that carries bi-directional fusion of the acquired pose spike features, allowing for the distribution of pose information especially to those in the early time. In our spiking transformer, the attention score between binary spike vectors is based on the normalized Hamming similarity, which, as shown in Proposition 1, amounts to the scaled dot-product similarity between the real valued vectors used in the standard transformer [200]. In the final step, 2D average pooling is applied to the spatiotemporally aggregated spike features, which is followed by a direct regression to output the parametric 3D human poses over time.

Our contributions can be summarized as follows:

- This work addresses a relatively new task of 3D human pose tracking solely based on events from an event camera.

- We propose an end-to-end SNNs approach, which specifically incorporates a novel Spiking Spatiotemporal Transformer module to tackle the one-directional temporal dependency issue. This allows for the propagation of pose-related information to facilitate pose estimation especially for the early time steps. Extensive empirical experiments demonstrate the superior performance of our approach over existing SOTA methods, including EventCap [227], EventHPE [266] and ANNs baselines [26], [98], [115].Extensive empirical experiments demonstrate the superior performance of our approach over existing state-of-the-art (SOTA) methods, utilizing merely around 20% of the FLOPs and 3% of the energy cost consumed by the ANNs. Additionally, our approach also outperforms SOTA SNNs baselines [57], [235], [262] in this regression task of human pose tracking.

- A large-scale dataset, SynEventHPD, is constructed for the task of event-based 3D human pose tracking. It consists of synthesized events data from multiple motion capture datasets, *i.e.*, Human3.6M [80], AMASS [122], PHSPD [270] and MMHPSD-Gray [266]. Consequently, it covers a variety of motions such as juggling, moon-walking, jumping rope, vaulting and scampering, with a total size of 45.72 hours event streams – more than 10 times larger than MMHPSD [266], the largest existing event-based pose tracking dataset. The details are summarized in Tab. 5.1, and empirical studies have showcased the usefulness of our new dataset. [1]

5.2 Preliminary Backgrounds

Spiking neuron model commonly refers to the leaky integrate and fire (LIF) model, a fundamental unit in SNNs. Its working process is shown in

[1] Our code and dataset are available at https://github.com/JimmyZou/HumanPoseTracking_SNN

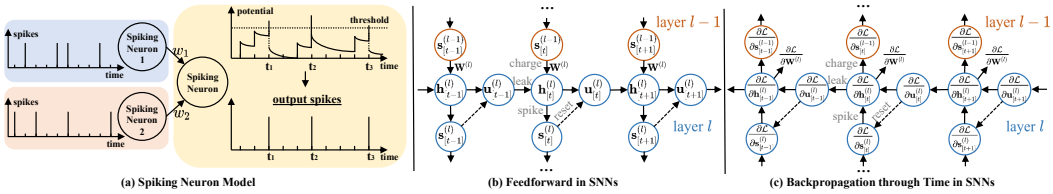


Figure 5.2: (a) Illustration of spiking neuron model. (b) Feedforward in SNNs. (c) Backpropagation Through Time in SNNs.

Fig. 5.2 (a). A LIF neuron maintains a membrane potential $u_{[t]}$ with a leaky constant τ, which may be modified only when new spiking trains $X_{[t]}$ are received from its connected neurons in T time steps. The neuron then outputs a spike $s_{[t]}$ and reset its potential by $V_{\text{th}} - u_{\text{rest}}$, if its potential exceeds a predetermined threshold, V_{th}, where soft reset [57] is adopted in our work. The model is formulated as follows:

$$h_{[t]} = u_{[t-1]} - \frac{1}{\tau}(u_{[t-1]} - u_{\text{rest}}) + X_{[t]}, \tag{5.1}$$

$$s_{[t]} = \Theta(h_{[t]} - V_{\text{th}}), \tag{5.2}$$

$$u_{[t]} = h_{[t]} - (V_{\text{th}} - u_{\text{rest}})s_{[t]}, \tag{5.3}$$

where Θ is the Heaviside step function,

$$\Theta(h_{[t]} - V_{\text{th}}) = \begin{cases} 1, & \text{if } h_{[t]} - V_{\text{th}} \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

Feedforward in SNNs consists of multiple layers of connected spiking neurons, shown in Fig. 5.2 (b). Assume there are $N^{(l)}$ neurons in the l-th layer, and use the vector forms of $\mathbf{u}_{[t]}^{(l)} \in \mathbb{R}^{N^{(l)}}$ and $\mathbf{s}_{[t]}^{(l)} \in \{0,1\}^{N^{(l)}}$ to represent their respective membrane potentials and output spikes at time step t. Let $\mathbf{W}^l \in \mathbb{R}^{N^{(l)} \times N^{(l-1)}}$ denote the connecting weights between layer $l-1$ and l, $\lambda = 1 - \frac{1}{\tau}$ be the leaky constant of LIF neuron model, and set u_{rest} to 0,

feedforward in SNNs becomes

$$\mathbf{h}_{[t]}^{(l)} = \underbrace{\lambda \mathbf{u}_{[t-1]}^{(l)}}_{\text{leak}} + \underbrace{\mathbf{W}^{(l)} \mathbf{s}_{[t]}^{(l-1)}}_{\text{charge}}, \tag{5.5}$$

$$\mathbf{s}_{[t]}^{(l)} = \underbrace{\Theta(\mathbf{h}_{[t]}^{(l)} - V_{\text{th}})}_{\text{spike}}, \tag{5.6}$$

$$\mathbf{u}_{[t]}^{(l)} = \mathbf{h}_{[t]}^{(l)} \underbrace{- V_{\text{th}} \mathbf{s}_{[t]}^{(l)}}_{\text{reset}}. \tag{5.7}$$

Computation and energy consumption of SNNs are often lower than ANNs, partly owing to the binary output of spiking neurons. According to previous works [235], [262], the l-th linear layer in the ANNs requires $\mathcal{O}(TN^{(l-1)}N^{(l)})$ FLOPs [2], measured in terms of *multiply-and-accumulate (MAC) operations*. In the context of SNNs, when assuming a spiking rate of ρ for l-th linear layer, as derived Eq. (5.5), it only requires $\mathcal{O}(\rho TN^{(l-1)}N^{(l)})$ FLOPs measured in terms of *accumulate (AC) operations*, where the computation of inactive neurons ($s_{[t]}^{(l-1)} = 0$) can be skipped. As for the energy consumption, we assume the data for various operations are 32-bit floating-point implementation in 45nm technology [77], in which $E_{\text{MAC}} = 4.6pJ$ and $E_{\text{AC}} = 0.9pJ$.

Backpropagation through time in SNNs is shown in Fig. 5.2 (c). Given the backpropagate gradients from the last layer $\frac{\partial \mathcal{L}}{\mathbf{s}_{[t]}^{(l)}}$, we can unfold the iterative update of membrane potential for T time steps and calculate the gradients $\frac{\partial \mathcal{L}}{\partial \mathbf{s}_{[t]}^{(l-1)}}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}}$ respectively. As is analysed in [223], since the derivative of the Heaviside step function is 0 almost everywhere, we can detach the neuron reset operation from the computational graph and do not backpropagating gradients in this path.

Considering only $\mathbf{s}_{[k]}^{(l)}$ $(k \geq t)$ depends on $\mathbf{s}_{[t]}^{(l-1)}$, the loss function can be described as

$$\mathcal{L}\Big(\mathbf{s}_{[t]}^{(l)}\Big(\mathbf{h}_{[t]}^{(l)}(\mathbf{s}_{[t]}^{(l-1)})\Big),\quad \mathbf{s}_{[t+1]}^{(l)}\Big(\mathbf{h}_{[t+1]}^{(l)}\Big(\mathbf{h}_{[t]}^{(l)}(\mathbf{s}_{[t]}^{(l-1)}), \mathbf{s}_{[t]}^{(l)}(\mathbf{h}_{[t]}^{(l)}(\mathbf{s}_{[t]}^{(l-1)}))\Big)\Big),\quad \mathbf{s}_{[t+2]}^{(l)}\Big(\ldots\Big), \ldots\Big). \tag{5.8}$$

Then we have the gradient of loss with respect to $\mathbf{s}_{[t]}^{(l-1)}$ from layer l to layer

[2] FLOPs refers to the number of floating-point operations.

$l-1$ as

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{s}_{[t]}^{(l-1)}} &= \frac{\partial \mathcal{L}}{\partial \mathbf{s}_{[t]}^{(l)}} \frac{\partial \mathbf{s}_{[t]}^{(l)}}{\partial \mathbf{h}_{[t]}^{(l)}} \frac{\partial \mathbf{h}_{[t]}^{(l)}}{\partial \mathbf{s}_{[t]}^{(l-1)}} + \frac{\partial \mathcal{L}}{\partial \mathbf{s}_{[t+1]}^{(l)}} \frac{\partial \mathbf{s}_{[t+1]}^{(l)}}{\partial \mathbf{h}_{[t+1]}^{(l)}} \Bigg(\underbrace{\frac{\partial \mathbf{h}_{[t+1]}^{(l)}}{\partial \mathbf{u}_{[t]}^{(l)}} \frac{\partial \mathbf{u}_{[t]}^{(l)}}{\partial \mathbf{h}_{[t]}^{(l)}}}_{\text{leak path}} \frac{\partial \mathbf{h}_{[t]}^{(l)}}{\partial \mathbf{s}_{[t]}^{(l-1)}} \\
&\quad + \underbrace{\frac{\partial \mathbf{h}_{[t+1]}^{(l)}}{\partial \mathbf{u}_{[t]}^{(l)}} \frac{\partial \mathbf{u}_{[t]}^{(l)}}{\partial \mathbf{s}_{[t]}^{(l)}} \frac{\partial \mathbf{s}_{[t]}^{(l)}}{\partial \mathbf{h}_{[t]}^{(l)}}}_{\text{reset path}} \frac{\partial \mathbf{h}_{[t]}^{(l)}}{\partial \mathbf{s}_{[t]}^{(l-1)}} \Bigg) + \dots \\
&= \frac{\partial \mathcal{L}}{\partial \mathbf{s}_{[t]}^{(l)}} \frac{\partial \mathbf{s}_{[t]}^{(l)}}{\partial \mathbf{h}_{[t]}^{(l)}} \frac{\partial \mathbf{h}_{[t]}^{(l)}}{\partial \mathbf{s}_{[t]}^{(l-1)}} + \frac{\partial \mathcal{L}}{\partial \mathbf{s}_{[t+1]}^{(l)}} \frac{\partial \mathbf{s}_{[t+1]}^{(l)}}{\partial \mathbf{h}_{[t+1]}^{(l)}} \Bigg(\underbrace{\lambda}_{\text{leak path}} - \underbrace{V_{\text{th}} \frac{\partial \mathbf{s}_{[t]}^{(l)}}{\partial \mathbf{h}_{[t]}^{(l)}}}_{\text{reset path}} \Bigg) \frac{\partial \mathbf{h}_{[t]}^{(l)}}{\partial \mathbf{s}_{[t]}^{(l-1)}} + \dots \\
&= \sum_{k=t}^{T} \underbrace{\frac{\partial \mathcal{L}}{\partial \mathbf{s}_{[k]}^{(l)}}}_{\substack{\text{gradient from} \\ \text{last layer}}} \underbrace{\frac{\partial \mathbf{s}_{[k]}^{(l)}}{\partial \mathbf{h}_{[k]}^{(l)}}}_{\substack{\text{surrogate} \\ \text{gradient}}} \Big(\mathbf{1} + \prod_{\tau=t-1}^{k-1} \big(\lambda - V_{\text{th}} \underbrace{\frac{\partial \mathbf{s}_{[\tau]}^{(l)}}{\partial \mathbf{h}_{[\tau]}^{(l)}}}_{\substack{\text{surrogate} \\ \text{gradient}}} \big) \Big) \mathbf{W}^{(l)} \\
&\overset{\substack{\text{detach} \\ \text{reset}}}{\approx} \sum_{k=t}^{T} \lambda^{k-t} \frac{\partial \mathcal{L}}{\partial \mathbf{s}_{[k]}^{(l)}} \frac{\partial \mathbf{s}_{[k]}^{(l)}}{\partial \mathbf{h}_{[k]}^{(l)}} \mathbf{W}^{(l)}.
\end{aligned}$$

The loss at time step t only depends on $\mathbf{h}_{[k]}^{(l)}$ where $k \le t$,

$$\begin{aligned}
\mathcal{L}_t\Big(\mathbf{h}_{[t]}^{(l)}\Big) &= \mathcal{L}_t\Bigg(\mathbf{h}_{[t]}^{(l)}\Big(\mathbf{W}^{(l)}, \mathbf{u}_{[t-1]}^{(l)}\Big)\Bigg) \\
&= \mathcal{L}_t\Bigg(\mathbf{h}_{[t]}^{(l)}\Big(\underbrace{\mathbf{W}^{(l)}}_{\text{charge path}}, \ \mathbf{u}_{[t-1]}^{(l)}\Big(\underbrace{\mathbf{h}_{[t-1]}^{(l)}(\mathbf{W}^{(l)})}_{\text{leak path}}, \underbrace{\mathbf{s}_{[t-1]}^{(l)}\big(\mathbf{h}_{[t-1]}^{(l)}(\mathbf{W}^{(l)})\big)}_{\text{spike and reset path}}\Big)\Big)\Bigg).
\end{aligned}$$

So we have the unfolded gradients of $\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}}$ as

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} &= \sum_{t=0}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} = \sum_{t=0}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{h}_{[t]}^{(l)}} \frac{\partial \mathbf{h}_{[t]}^{(l)}}{\partial \mathbf{W}^{(l)}}, \\
&= \sum_{t=0}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{h}_{[t]}^{(l)}} \Big(\underbrace{\frac{\partial \mathbf{h}_{[t]}^{(l)}}{\partial \mathbf{W}^{(l)}}}_{\text{charge path}} + \underbrace{\frac{\partial \mathbf{h}_{[t]}^{(l)}}{\partial \mathbf{u}_{[t-1]}^{(l)}} \frac{\partial \mathbf{u}_{[t-1]}^{(l)}}{\partial \mathbf{h}_{[t-1]}^{(l)}} \frac{\partial \mathbf{h}_{[t-1]}^{(l)}}{\partial \mathbf{W}^{(l)}}}_{\text{leak path}} + \underbrace{\frac{\partial \mathbf{h}_{[t]}^{(l)}}{\partial \mathbf{u}_{[t-1]}^{(l)}} \frac{\partial \mathbf{u}_{[t-1]}^{(l)}}{\partial \mathbf{s}_{[t-1]}^{(l)}} \frac{\partial \mathbf{s}_{[t-1]}^{(l)}}{\partial \mathbf{h}_{[t-1]}^{(l)}} \frac{\partial \mathbf{h}_{[t-1]}^{(l)}}{\partial \mathbf{W}^{(l)}}}_{\text{spike and reset path}} \Big), \\
&= \sum_{t=0}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{h}_{[t]}^{(l)}} \Big(\mathbf{s}_{[t]}^{(l)} + \lambda \frac{\partial \mathbf{h}_{[t-1]}^{(l)}}{\partial \mathbf{W}^{(l)}} - \lambda V_{\text{th}} \frac{\partial \mathbf{s}_{[t-1]}^{(l)}}{\partial \mathbf{h}_{[t-1]}^{(l)}} \frac{\partial \mathbf{h}_{[t-1]}^{(l)}}{\partial \mathbf{W}^{(l)}} \Big), \\
&= \sum_{t=0}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{h}_{[t]}^{(l)}} \Big(\mathbf{s}_{[t]}^{(l)} + \lambda (1 - V_{\text{th}} \frac{\partial \mathbf{s}_{[t-1]}^{(l)}}{\partial \mathbf{h}_{[t-1]}^{(l)}}) \underbrace{\frac{\partial \mathbf{u}_{[t-1]}^{(l)}}{\partial \mathbf{W}^{(l)}}}_{\substack{\text{unroll} \\ \text{over time}}} \Big), \\
&= \sum_{t=0}^{T} \underbrace{\frac{\partial \mathcal{L}}{\partial \mathbf{s}_{[t]}^{(l)}}}_{\substack{\text{gradient from} \\ \text{last layer}}} \underbrace{\frac{\partial \mathbf{s}_{[t]}^{(l)}}{\partial \mathbf{h}_{[t]}^{(l)}}}_{\substack{\text{surrogate} \\ \text{gradient}}} \Big(\mathbf{s}_{[t]}^{(l)} + \sum_{k=0}^{t-1} \Big(\prod_{\tau=k}^{t-1} \lambda (1 - V_{\text{th}} \underbrace{\frac{\partial \mathbf{s}_{[\tau]}^{(l)}}{\partial \mathbf{h}_{[\tau]}^{(l)}}}_{\substack{\text{surrogate} \\ \text{gradient}}}) \Big) \mathbf{s}_{[k]}^{(l)} \Big) \\
&\overset{\substack{\text{detach} \\ \text{reset}}}{\approx} \sum_{t=0}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{s}_{[t]}^{(l)}} \frac{\partial \mathbf{s}_{[t]}^{(l)}}{\partial \mathbf{h}_{[t]}^{(l)}} \Big(\sum_{k=0}^{t} \lambda^{t-k} \mathbf{s}_{[k]}^{(l)} \Big).
\end{aligned}
$$

Training SNNs from scratch is difficult mainly due to the non-differentiable property of Heaviside step function and the problem of gradient vanishing. Existing efforts summarized in [107] solve it by using surrogate derivatives to approximate the gradients of Heaviside step function. Following [57], the surrogate gradient function we used in this work is

$$
\frac{\partial s_{[t]}}{\partial h_{[t]}} = \begin{cases} \frac{c}{2(1+(\frac{\pi}{2} c (h_{[t]} - V_{\text{th}}))^2)}, & \text{if } s_{[t]} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{5.9}
$$

where c is the hyper-parameter to control the smoothness of the surrogate gradients.

Scaled dot-product attention is defined as

$$
\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}})\mathbf{V}, \tag{5.10}
$$

in the standard transformer [200], where $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{N \times d_k}$ are queries and keys and $\mathbf{V} \in \mathbb{R}^{N \times d_v}$ is values, with N being the length of input sequence, d_k the dimension of a single query $\mathbf{q}$ or key $\mathbf{k}$, and d_v the dimension of a single value $\mathbf{v}$. The scaling factor of $1/\sqrt{d_k}$ is to normalize the dot-product $\mathbf{q}\mathbf{k}^\top$ to have

mean 0 and variance 1, assuming the components of $\mathbf{q}$ and $\mathbf{k}$ are independent variables with mean 0 and variance 1.

5.3 Our approach

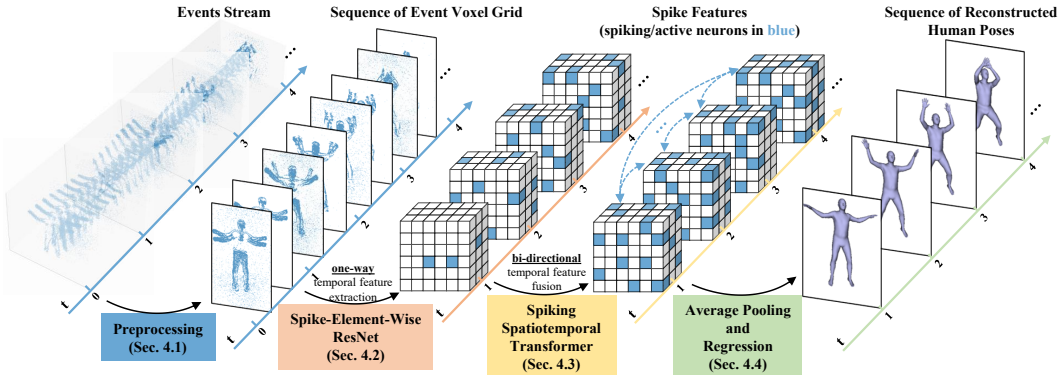


Figure 5.3: Pipeline of our sparse deep learning approach.

As summarized in Fig. 5.3, our approach consists of four main sections. 1) Preprcessing in Sec. 5.3.1, is to convert an stream of event into a sequence of event voxel grids [59] with the same time interval. 2) SEW-ResNet [57] in Sec. 5.3.2, is employed as the backbone to extract pose spike features from the input sequence of event voxel. 3) Since SEW-ResNet only considers the one-directional temporal relationship, we propose a novel Spiking Spatiotemporal Transformer for the bi-directional fusion of pose spike features in Sec. 5.3.3, allowing for the compensation of missing pose information, especially for the early time steps. 4) The final stage, illustrated in Sec. 5.3.4, is to apply average pooling to the spatiotemporally aggregated spike features and then regress the parametric poses over time. In this work, a stream of events is the sole source of input, thus eliminating the reliance on gray-scale input images as in [227] or a prior knowledge of the starting pose as in [266]. Furthermore, our model is completely built upon SNNs instead of traditional ANNs or mixed architecture.

5.3.1 Preprocessing

Instead of a sequence of RGB frames captured by an RGB camera, an asynchronous stream of independent event signals is assembled by an event cam-

era as the input signals. This event stream is decomposed into a sequence of T packets of events, with each packet spanning the same length of time, $\mathcal{E} = \{\mathcal{E}_{[t]}\}_{t=1}^{T}$. Here, t indexes a specific event packet in the sequence. Following [62], [266], an event packet, $\mathcal{E}_{[t]}$, is represented as a voxel grid $H \times W \times C$ with each voxel corresponding to a particular spatial and temporal interval. The voxel value will be 1 if the number of events within the voxel is larger than a preset threshold, and 0 otherwise. This representation better preserves the temporal information of events, rather than collapsing them onto a single frame as mentioned in [59]. The processed sequence of event voxel grids is then fed into SNNs as input, denoted as $\mathbf{S}^{\text{in}} \in \{0,1\}^{T\times H\times W\times C}$. [3]

5.3.2 Spike-Element-Wise Residual Networks

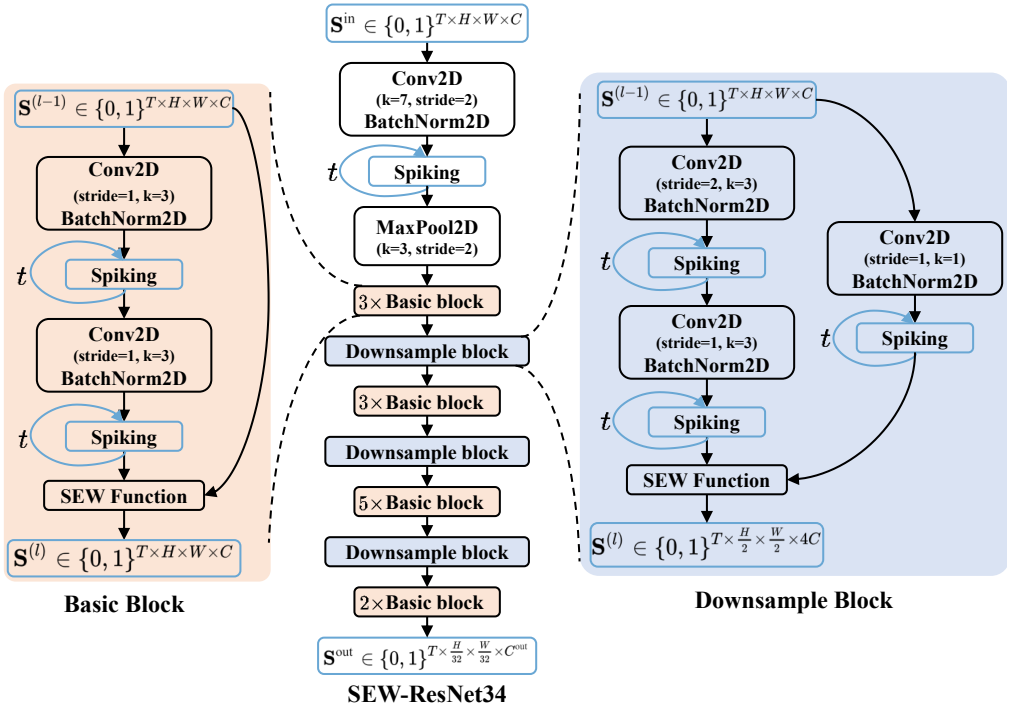


Figure 5.4: Architecture of SEW-ResNet34.

SEW-ResNet, proposed in [57], ranks among the most popular SNNs architectures. Originated from ResNet [73], significant differences are made in the redesign of identity mapping for SNNs using the *SEW Function*, which applies

[3] For clarity, we will generally refer to the size of input spike tensor for different blocks or modules as $T \times H \times W \times C$ in subsequent sections.

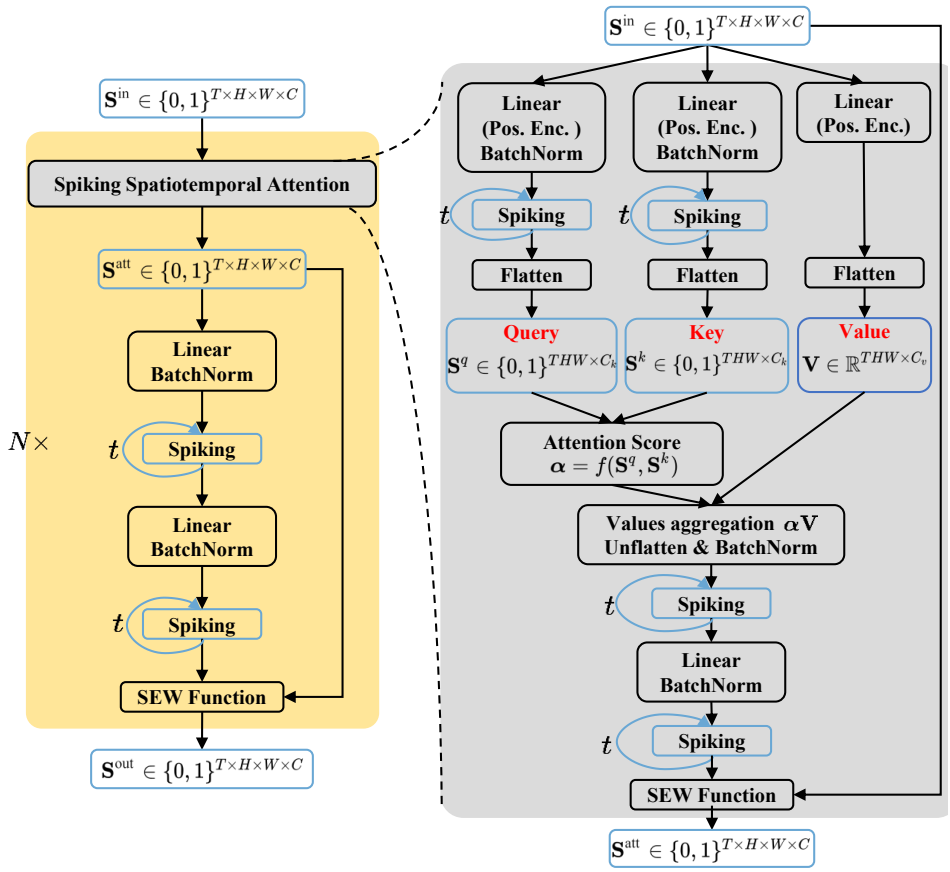
element-wise addition to spike tensors rather than the pre-spiking membrane potentials. This design not only establishes the identity mapping of residual learning in SNNs, but also addresses the vanishing or exploding gradient issue. In our pipeline, it is incorporated as the SNNsbackbone to extract spike pose features. The spatial size of the output is 1/32 of the input with a channel size of 512 for SEW-ResNet18, 34 and 2048 for SEW-ResNet50, 101, 152. The detailed architecture of SEW-ResNet34 is presented in Fig. 5.4.

SEW-ResNet consists of two types of blocks: the downsample block and the basic block. The downsample block normally reduces the spatial size of input spike tensor by 2 and expands the channel size by 4 through convolutional layers, while the basic block keeps the size of input spike tensor unchanged for residual learning. The final layer in both types of blocks is element-wise identity mapping via *SEW Function*, where spike-element-wise functions between two input spike tensors are applied, such as ADD, AND or IAND. Given the input spike tensor $\mathbf{S}^{\text{in}} \in \{0,1\}^{T\times H\times W\times C}$, the output spike tensor is assumed to be $\mathbf{S}^{\text{out}} \in \{0,1\}^{T\times \frac{H}{32}\times \frac{W}{32}\times C^{\text{out}}}$ where $C^{\text{out}} = 512$ for SEW-ResNet18 and 34, $C^{\text{out}} = 2048$ for SEW-ResNet50, 101 and 152.

5.3.3 Spiking Spatiotemporal Transformer

Spiking Spatiotemporal Transformer is shown in Fig. 5.5 (a). Given the input $\mathbf{S}^{\text{in}} \in \{0,1\}^{T\times H\times W\times C}$, the first step is to apply spiking spatiotemporal attention to combine bidirectional space-time features. A more comprehensive explanation of the attention module will be given later. It is followed by two linear spiking layer with batch normalization, also known as Feed-Forward Network (FFN) in the standard transformer [200]. The final step in the module is to apply SEW Function to the output of FFN and input spike tensor for residual learning. Then we get the output $\mathbf{S}^{\text{out}} \in \{0,1\}^{T\times H\times W\times C}$. This entire module can be stacked by N layers similar to the standard transformer [200].

Spiking Spatiotemporal Attention is illustrated in Fig. 5.5 (b). This module introduces self-attention that spans the entire space-time domain of spike tensors, effectively addressing the issue of one-directional temporal dependency flow in spiking layers of SNNs. Specifically, starting with the in-



(a) Spiking Spatiotemporal Transformer **(b) Spiking Spatiotemporal Attention**

Figure 5.5: (a) Architecture of our Spiking Spatiotemporal Transformer. (b) Architecture of Spiking Spatiotemporal Attention.

put spike tensor $\mathbf{S}^{\text{in}} \in \{0,1\}^{T\times H\times W\times C}$, we use two linear spiking layers with positional encodings and batch normalization to map the channel size from C to C_k. Subsequently, we flatten it across the spatiotemporal dimensions $T \times H \times W$ to obtain the spike query and key tensors, denoted as $\mathbf{S}^q, \mathbf{S}^k \in \{0,1\}^{THW\times C_k}$. Similarly, we obtain the real-valued tensor $\mathbf{V} \in \mathbb{R}^{THW\times C_v}$ by applying a non-spiking linear layer and positional encodings to transform the channel size from C to C_k and then flattening it across the spatiotemporal dimensions. The rationale for utilizing a non-spiking layer is to delay spiking until after feature aggregation has been achieved through self-attention. Next, the similarity between the spiking queries and keys is calculated using the function $\boldsymbol{\alpha} = f(\mathbf{S}^q, \mathbf{S}^k)$, which serves as the attention scores for values aggregation, $\boldsymbol{\alpha}\mathbf{V}$. The details of $f(\cdot)$ will be covered later. The aggregated

value tensor is then unflattened to be $\mathbb{R}^{T\times H\times W\times C_v}$, followed with a batch normalization and spiking layer. Afterwards, we use a spiking linear layer with batch normalization to map the channel size from C_v back to C. Finally, the SEW Function is applied to the attention output and the input spike tensor for residual learning, resulting in the output as $\mathbf{S}^{\text{att}} \in \{0,1\}^{T\times H\times W\times C}$. It is important to note that our model also supports multi-head attention.

Positional encodings are added in the first layer of the spiking spatiotemporal attention module, aiming to make the model aware of ordinal information in the input sequence. As the input of the attention module is binary spike tensor while positional encodings are float, direct addition would violate the fast computation in SNNs. So we add the encodings after the linear layer, but before the batch normalization and spiking layer. Besides, as the spiking layer generates spikes by rolling over T time steps, we scale the positional encodings by $1/T$ to maintain consistency across models with varying T. The definition mostly follows [200] as

$$\begin{aligned}\text{PE}(\text{pos}, 2i) &= \frac{1}{T}\sin(\text{pos}/10000^{2i/C_k}),\\ \text{PE}(\text{pos}, 2i+1) &= \frac{1}{T}\cos(\text{pos}/10000^{2i/C_k}),\end{aligned}$$

where pos represents the position in the sequence, while $2i$ or $2i+1$ denotes the position of C_k channel.

Attention scores in both the standard transformer [200] and the recently introduced spiking transformer [262] are commonly computed using dot-products. However, this approach is not well-defined for binary spike vectors. *When there are zero components in the spike key vector, the dot-product will invariably disregard the values of corresponding components in the spike query vector.* As an example, consider two spike query vectors, $\mathbf{s}_1^q$ and $\mathbf{s}_2^q$, which differ only in the c-th element such that $\mathbf{s}_1^q[c] = 0$, $\mathbf{s}_2^q[c] = 1$. For a spike key vector with its c-th element equal to 0 ($\mathbf{s}^k[c] = 0$), the dot-product will always yield the same attention score for the two different spike queries: $\mathbf{s}^{k\top}\cdot\mathbf{s}_1^q = \mathbf{s}^{k\top}\cdot\mathbf{s}_2^q$. Only when the c-th element is equal to 1 ($\mathbf{s}^k[c] = 1$), the spike key vector can differentiate between these two different queries. This means that the dot-product used in [200], [262] is actually unable to precisely

measure the similarity between two binary spike vectors.

Lemma 1 (Johnson–Lindenstrauss Lemma on Binary Embedding [81], [237])**.** *Let* $\{\mathbf{x}_i\}_{i=1}^{M}$ *be set of* M *real-valued points, define its one bit quantization of the projections,*

$$\mathbf{s}(\mathbf{x}) = sign(\mathbf{A}\mathbf{x}),$$

where $\mathbf{s}(\mathbf{x}) \in \{0,1\}^{C_k}$ *is the binary embedding of* $\mathbf{x} \in \mathbb{R}^{d_k}$ *and* $\mathbf{A} \in \mathbb{R}^{C_k \times d_k}$ *is a projection matrix with each entry generated independently from the normal distribution* $\mathcal{N}(0,1)$*. Given* $C_k > \frac{\log M}{\delta^2}$*, for any two points among* M*,*

$$|f_{\mathcal{H}}(\mathbf{s}_i, \mathbf{s}_j) - f_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j)| \leq \delta \tag{5.11}$$

holds true with probability at least $1 - 2e^{-\delta^2 C_k}$*. Here* $f_{\mathcal{H}}$ *is the normalized Hamming distance defined as*

$$f_{\mathcal{H}}(\mathbf{s}_i, \mathbf{s}_j) = \frac{1}{C_k} \sum_{k=1}^{C_k} \mathbb{1}(\mathbf{s}_{ik} \neq \mathbf{s}_{jk}), \tag{5.12}$$

and $f_{\mathcal{C}}$ *is cosine distance defined as*

$$f_{\mathcal{C}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\pi} \arccos\Big(\frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}\Big). \tag{5.13}$$

Proposition 1. *Let* $\mathbf{q}_i, \mathbf{k}_j \in \mathbb{R}^{d_k}$ *be a single query and key of real-valued points in the standard transformer. Define* $\mathbf{s}_i^q, \mathbf{s}_j^k \in \{0,1\}^{C_k}$ *as the corresponding binary embedding defined as*

$$\mathbf{s}_i^q(\mathbf{q}_i) = sign(\mathbf{A}\mathbf{q}_i), \quad \mathbf{s}_j^k(\mathbf{k}_j) = sign(\mathbf{A}\mathbf{k}_j),$$

where $\mathbf{A} \in \mathbb{R}^{C_k \times d_k}$ *is a projection matrix with each entry generated independently from the normal distribution* $\mathcal{N}(0,1)$*. Given that* $C_k > \frac{\log M}{\delta^2}$*, we have*

$$g(d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) - \delta) \leq d_{\mathcal{C}}(\mathbf{q}_i, \mathbf{k}_j) \leq g(d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) + \delta), \tag{5.14}$$

with probability at least $1 - 2e^{-\delta^2 C_k}$*. Here* $g(x) = \cos(\pi(1-x))$ *is a continuous and monotone function for* $x \in [0,1]$*,* M *is the number of all possible keys and queries given by the finite training set,* $d_{\mathcal{H}} \in [0,1]$ *is the normalized Hamming similarity defined as*

$$d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) = 1 - \frac{1}{C_k} \sum_{c=1}^{C_k} \mathbb{1}(\mathbf{s}_{ic}^q \neq \mathbf{s}_{jc}^k), \tag{5.15}$$

and $d_{\mathcal{C}} \in [0,1]$ *is cosine similarity defined as*

$$d_{\mathcal{C}}(\mathbf{q}_i, \mathbf{k}_j) = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\|\mathbf{q}_i\|\|\mathbf{k}_j\|}. \tag{5.16}$$

Proof. By substituting $\mathbf{x}_i, \mathbf{x}_j, \mathbf{s}_i, \mathbf{s}_j$ in Eq. (5.11) with $\mathbf{q}_i, \mathbf{k}_j, \mathbf{s}_i^q, \mathbf{s}_j^k$ respectively, we have

$$f_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) - \delta \leq f_{\mathcal{C}}(\mathbf{q}_i, \mathbf{k}_j) \leq f_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) + \delta.$$

After replacing $f_{\mathcal{H}}, f_{\mathcal{C}}$ defined in Eq. (5.12) and (5.13) with $d_{\mathcal{H}}, d_{\mathcal{C}}$ defined in Eq. (5.15) and (5.16), we have

$$1 - d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) - \delta \leq \frac{1}{\pi} \arccos\Big(d_{\mathcal{C}}(\mathbf{q}_i, \mathbf{k}_j)\Big) \leq 1 - d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) + \delta$$
$$\cos\Big(\pi\big(1 - d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) + \delta\big)\Big) \leq d_{\mathcal{C}}(\mathbf{q}_i, \mathbf{k}_j) \leq \cos\Big(\pi\big(1 - d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) - \delta\big)\Big).$$

Define the function $g(x) = \cos(\pi(1-x))$, we have

$$g(d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) - \delta) \leq d_{\mathcal{C}}(\mathbf{q}_i, \mathbf{k}_j) \leq g(d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) + \delta).$$

□

Proposition 1 reveals that cosine similarity $d_{\mathcal{C}}$ between real-valued queries and keys is bounded within $[g(d_{\mathcal{H}} - \delta), g(d_{\mathcal{H}} + \delta)]$, where $d_{\mathcal{H}}$ is the normalized Hamming similarity between corresponding binary spike queries and keys. When the channel size C_k is large enough, $d_{\mathcal{C}}$ approximates $g(d_{\mathcal{H}})$ with high probability. Given that g is a continuous and monotonic function for $d_{\mathcal{H}} \in [0, 1]$, we propose a direct utilization of $d_{\mathcal{H}}$ to compute the attention scores between binary spike queries and keys in our Spiking Spatiotemporal Transformer, which imitates the scaled dot-product similarity for real-valued vectors in the standard transformer [200].

The gradient of normalized Hamming similarity does not exist since Eq. (5.15) is a non-differentiable function. Thus we approximate $d_{\mathcal{H}}$ by

$$d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k) \approx 1 - \frac{1}{C_k} \sum_{c=1}^{C_k} \left[\mathbf{s}_{ic}^q \cdot (1 - \mathbf{s}_{jc}^k) + (1 - \mathbf{s}_{ic}^q) \cdot \mathbf{s}_{jc}^k\right]. \tag{5.17}$$

As a result, the approximate gradients of normalized Hamming similarity function are given by:

$$\frac{\partial d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k)}{\partial \mathbf{s}_i^q} \approx \frac{2\mathbf{s}_j^k - 1}{C_k}, \quad \frac{\partial d_{\mathcal{H}}(\mathbf{s}_i^q, \mathbf{s}_j^k)}{\partial \mathbf{s}_j^k} \approx \frac{2\mathbf{s}_i^q - 1}{C_k}.$$

5.3.4 Parametric Pose and Shape Regression

Parametric human pose and shape used in this work is SMPL model [116]. Given the shape parameters $\boldsymbol{\beta}$, pose parameters $\boldsymbol{\theta}$ and global translations $\mathbf{d}$, the model outputs a triangular mesh with 6,890 vertices at each time step, that is $\mathcal{M}(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{d}) \in \mathbb{R}^{T\times 6890\times 3}$ for T time steps in total. The shape parameters at time step t, denoted as $\boldsymbol{\beta}_{[t]} \in \mathbb{R}^{1\times 10}$, are linear coefficients of PCA shape space, learned from a large number of registered body scans. These parameters mainly describe individual body features such height, weight and body proportions. The pose parameters at time step t, denoted as $\boldsymbol{\theta}_{[t]} \in \mathbb{R}^{1\times 72}$, represent the articulated poses of the triangular mesh, consisting of a global rotation and relative rotations of the 24 joints in axis-angle form. The global translations of human body at time step t is denoted by $\mathbf{d}_{[t]} \in \mathbb{R}^{1\times 3}$. To produce the final parametric shapes, the template body is deformed using shape- and pose-dependent deformations, articulated through forward kinematics to its target pose, and further transformed through linear blend skinning and global translation. Meanwhile, the 3D and 2D joint positions, denoted as $\mathbf{J}_{3\mathrm{D}}$ and $\mathbf{J}_{2\mathrm{D}}$, are obtained by regressing from the output vertices and projecting the 3D joints onto the 2D images.

We show the process in Fig. 5.6 where we apply the 2D average pooling to the input spike tensor and then directly regress the shape parameters $\hat{\boldsymbol{\beta}}$, pose parameters $\hat{\boldsymbol{\theta}}$ and global translations $\hat{\mathbf{d}}$ via three linear layers in parallel. Based on the predicted parameters, we obtain the corresponding parametric shapes and joint positions $\hat{\mathbf{J}}_{3\mathrm{D}}, \hat{\mathbf{J}}_{2\mathrm{D}}$ by SMPL model across T time steps. When projecting 3D joints on 2D images, as the global translations under the camera coordinate are predicted, we can use predefined camera intrinsic parameters to reduce the redundancy of prediction.

The training losses for our model are introduced as follows:

$$\mathcal{L} = \lambda_{\mathrm{pose}}\mathcal{L}_{\mathrm{pose}} + \lambda_{\mathrm{shape}}\mathcal{L}_{\mathrm{shape}} + \lambda_{\mathrm{trans}}\mathcal{L}_{\mathrm{trans}} + \lambda_{3\mathrm{D}}\mathcal{L}_{3\mathrm{D}} + \lambda_{2\mathrm{D}}\mathcal{L}_{2\mathrm{D}},$$

where λ_{pose}, λ_{shape}, λ_{trans}, $\lambda_{3\mathrm{D}}$ and $\lambda_{2\mathrm{D}}$ are the corresponding loss weights. For the poses loss, we use the 6D representation of rotations, which has been shown to perform better than the 3D axis-angle representation in [261], [266]

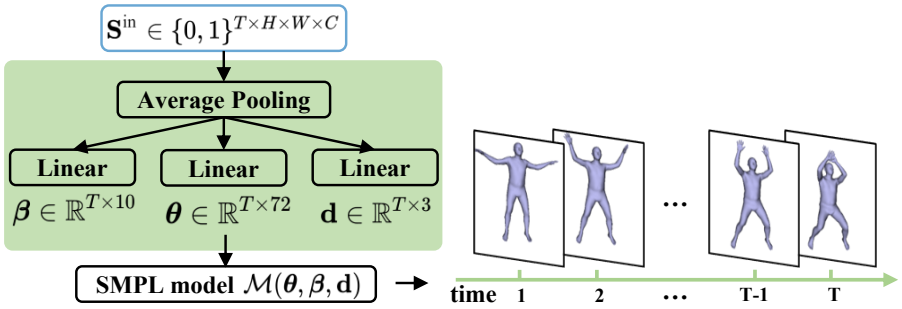


Figure 5.6: Human poses and shapes regression.

for human pose estimation. Then we use the geodesic distance in $SO(3)$ to measure the distance between the predicted and target poses:

$$\mathcal{L}_{\text{pose}} = \sum_{t=1}^{T}\sum_{j=1}^{24} \arccos^2\Big(\frac{\text{Tr}\left(R^{\top}(\boldsymbol{\theta}^j_{[t]})R(\hat{\boldsymbol{\theta}}^j_{[t]})\right)-1}{2}\Big), \tag{5.18}$$

where $R(\cdot)$ is the function that transforms the 6D rotational representation to the 3×3 rotation matrix and j is the joint index. Other losses are basically Euclidean distances between the predicted and target values as follows:

$$\mathcal{L}_{\text{shape}} = \sum_{t=1}^{T} \|\boldsymbol{\beta}_{[t]} - \hat{\boldsymbol{\beta}}_{[t]}\|^2,$$

$$\mathcal{L}_{\text{trans}} = \sum_{t=1}^{T} \|\mathbf{d}_{[t]} - \hat{\mathbf{d}}_{[t]}\|^2,$$

$$\mathcal{L}_{\text{3D}} = \sum_{t=1}^{T}\sum_{j=1}^{24} \|\mathbf{J}^j_{\text{3D}[t]} - \hat{\mathbf{J}}^j_{\text{3D}[t]}\|^2,$$

$$\mathcal{L}_{\text{2D}} = \sum_{t=1}^{T}\sum_{j=1}^{24} \|\mathbf{J}^j_{\text{2D}[t]} - \hat{\mathbf{J}}^j_{\text{2D}[t]}\|^2.$$

5.3.5 Our SynEventHPD Dataset

Currently, the largest event-based dataset for human pose estimation is MMHPSD, which includes 15 subjects, 21 different actions and a total of 4.39 hours of event streams [266]. However, this dataset's limited variety of motions restricts the generalization ability of trained models. To address this issue, we propose to synthesize event data from multiple motion capture datasets, including Human3.6M [80], AMASS [122], PHSPD [270] and MMHPSD-Gray [266],

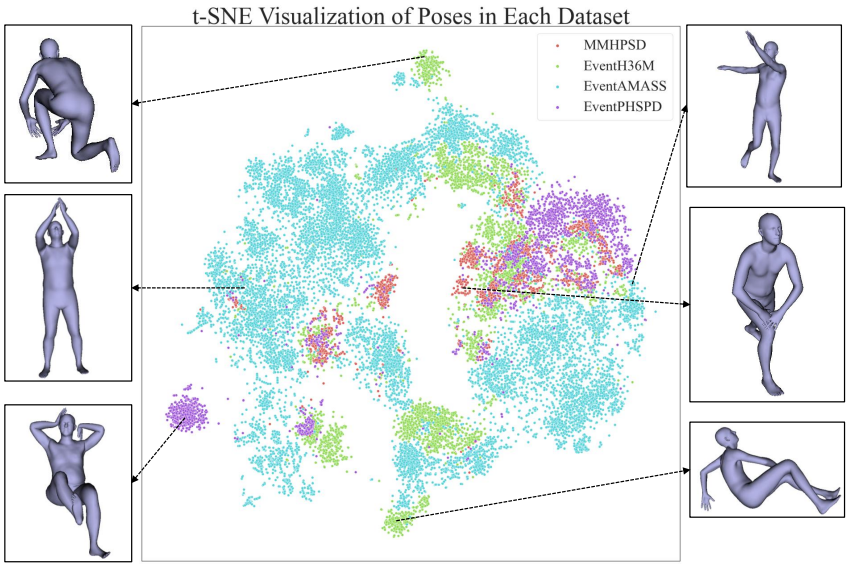


Figure 5.7: t-SNE visualization of poses from each sub-dataset in our SynEventHPD dataset.

Dataset	R/S	Sub #	Str #	Len (hrs)	AvgLen (mins)	Pose
MMHPSD [266]	Real	15	178	4.39	1.48	✓
EventH36M	Syn	7	835	12.46	0.90	✓
EventAMASS	Syn	13	8028	23.54	0.18	✓
EventPHSPD	Syn	12	156	5.33	2.05	✓
SynMMHPSD	Syn	15	178	4.39	1.48	✓
SynEventHPD (Total)	Syn	47	9197	45.72	0.30	✓

Table 5.1: Summary of event-based datasets for 3D human pose tracking, including existing MMHPSD dataset and 4 sub-datasets in our SynEventHPD dataset.

to construct a large-scale synthetic dataset. Our synthetic dataset, called SynEventHPD, is a meta dataset consisting of 4 sub-datasets: EventH36M, EventAMASS, EventPHSPD and SynMMHPSD. In total, it contains 45.72 hours of event streams, which is more than 10 times the size of MMHPSD. The distribution of poses across all these datasets are visualized in Fig. 5.7 to highlight the variety. Other details are summarized in Tab. 5.1.

The synthesizing process of our new dataset is mainly based on the workflow proposed in [61]. Given an RGB video, we first detect the bounding box of the person in each frame using 2D pose annotations or 2D pose detector like OpenPose [24]. After calculating the global bounding box, we

crop each RGB frame accordingly and resize to 512×512 to maintain a same image size across different sub-datasets. Next, we apply the approach proposed in [84] for frame interpolation guided by the predicted optical flows to increase the frame rate of provided videos. After converting the high frame rate RGB videos to gray-scale images, we generate events by checking the brightness change at each pixel over time, where the contrast thresholds for positive and negative events are 0.3 and the minimum waiting period before a pixel can trigger a new event is 1e-4 seconds. This process is straightforward for Human3.6M [80], PHSPD [270] and MMHPSD [266], as they contain RGB or gray-scale videos. We name the three sub-datasets as EventH36M, EventPHSPD and SynMMHPSD respectively. Since the three datasets provide corresponding SMPL pose and shape annotations, we keep them in our dataset while calculating an optimal global translation for each frame by aligning projected 3D poses with annotated 2D poses on the image, using the default camera intrinsic $(\text{focal\_length}, \text{center}) = (671.72, 256.4)$. The FPS of pose annotations in EventPHSPD and SynMMHPSD is 15 while FPS in EventH36M is 10 based on the observations that the motions in Human3.6M are relatively slow.

AMASS [122] dataset only contains motion capture data without any RGB videos. In this regard, for each motion capture sequence, an avatar is randomly picked from 13 different avatars shown in Fig. 5.8, animated and rendered to form its corresponding RGB videos of size 512×512, obtained using one of the 4 predefined lightning conditions displayed in Tab. 5.2. These 4 lightning conditions represent the positions of top center, left and right top, left and right bottom with different strength of illumination ranging from 0 to 1. The following process of events generation is the same with the 3 sub-datasets mentioned above. As for the SMPL pose and shape annotations, the additional preprocessing step for AMASS is to properly scale the trajectory of each sequence to avoid out of camera scope. We observe the FPS of AMASS motion sequences contains 60, 100 and 120. To make FPS of annotations across all the sequences, we downsample the FPS of motion to 20 before animation. Finally, we obtain the sub-dataset EventAMASS.

Figure 5.8: Front and back views of 13 avatars used in EventAMASS dataset.

Lightning	(Position, Color)
1	([0, 0, -300], [1.0, 1.0, 1.0])
2	([-300, -300, -300], [0.8, 0.8, 0.8]) ([300, -300, -300], [0.8, 0.8, 0.8])
3	([300, 0, -300], [1, 1, 1]) ([-300, 0, -300], [0.4, 0.4, 0.4])
4	([300, 0, -300], [0.4, 0.4, 0.4]) ([-300, 0, -300], [1, 1, 1])

Table 5.2: Four predefined lightning conditions used for rendering in EventAMASS dataset.

Annotations provided in our dataset include pose and shape parameters of SMPL model, corresponding 2D/3D joint positions and the global translation under the default camera intrinsic parameters. We demonstrate the effectiveness of our large-scale synthetic dataset by showing four examples of images, event frames, and annotated poses in Fig. 5.9. We also show a motion clip from each sub-dataset in Fig. 5.10 to illustrate the efficacy of our synthetic events.

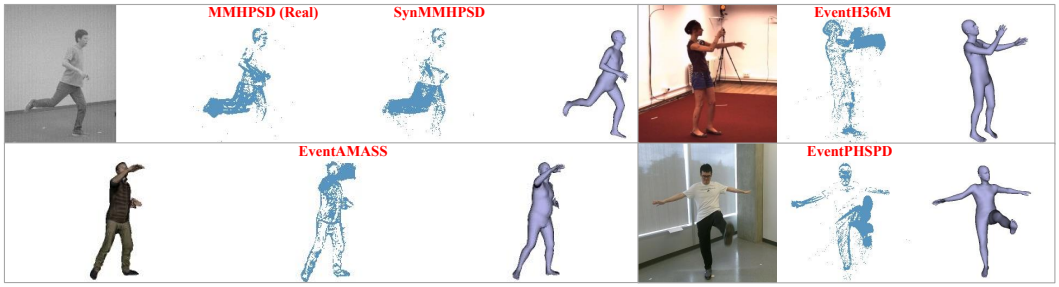


Figure 5.9: Sample examples of the synthesized event signals.

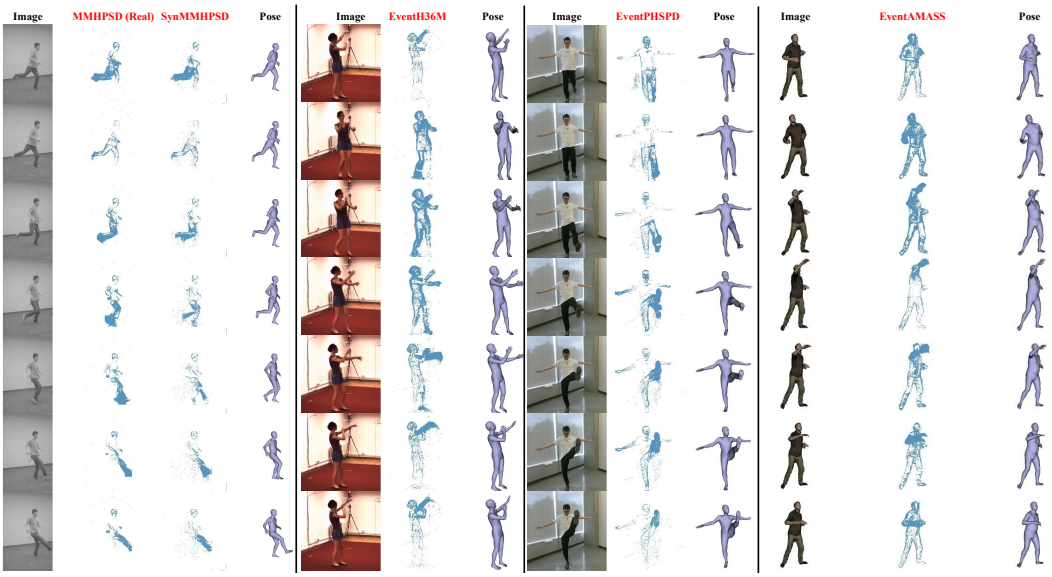


Figure 5.10: Example sequence from each sub-dataset in our SynEventHPD dataset.

5.4 Experiments

5.4.1 Empirical Results on MMHPSD Dataset

In this section, we start by outlining the implementation details for training and explaining the reported evaluation metrics. Subsequently, we compare our method with recent video-based and event-based human pose estimation approaches, emphasizing the competence of event signals for human pose tracking. We also compare our SNNs model with three popular ANNs models, illustrating the efficiency and effectiveness of our SNNs approach. Moreover, we contrast our approach with five recently proposed SNNs models, showcasing the superiority of our Spiking Spatiotemporal Transformer for bi-directional

Model	ANN/ SNN	Architecture
VIBE [98]	ANN	ResNet50 + GRU-1024hidden-bidirectional
MPS [220]		ResNet50 + Temporal Attentive Module
EventHPE [266]		ResNet50 + GRU-1024hidden-bidirectional
ResNet-GRU [98]		ResNet50 + GRU-1layer-1024hidden-bidirectional
ResNet-TF [26]		ResNet50 + Transformer-Encoder-2layers-384hidden-8heads + Transformer-Decoder-2layers-384hidden-8heads
Video-Swin [115]		Swin_Tiny-96hidden-depths=[2,2,6,2]-heads=[3,6,12,24]-window_size=(8,8,8)
SEW-ResNet-TF	Mix	SEW-ResNet50 + Transformer-Encoder-2layers-384hidden-8heads + Transformer-Decoder-2layers-384hidden-8heads
MA-SNN [235]	SNN	SEW-ResNet50 + Multi-dimensional Attention
SpikeFormer [262]		16x16patch-10layers-512hidden
Ours		ResNet50 + SpikingSpatiotemporal Transformer-2layers-1024-1head

Table 5.3: Architecture of different baseline models.

temporal information fusion in human pose tracking.

Implementation Details. For a fair comparison with prior works [227], [266], we follow the train and test set split for the MMHPSD dataset from [266], where subject 1, 2 and 7 are designated for testing and the remaining 12 subjects for training. We present the results of model trained with $T = 8$ time steps and $T = 64$ time steps. For event stream preprocessing, we convert each event packet into a voxel grid of size $256 \times 256 \times 4$. Empirically, we find that $C = 4$ is the best choice, as higher values do not show performance improvements in the ablation study. To fairly compare with other baselines in terms of the number of parameters, we use SEW-ResNet50 [57] as the backbone and configure the Spiking Spatiotemporal Transformer with 1024 hidden dimension, 1 attention head and 2 layers, resulting in 47.7M parameters. The architecture details of other baseline models are displayed in Tab. 5.3.

During training, to ensure robustness against both fast and slow motions, we augment the training samples in two ways: randomly selecting event stream of (0.5, 1, 2, 3) seconds for $T = 8$ and (4, 8, 16, 32) seconds for $T = 64$ as the input, spatially rotating the voxel grid with a random degree between -20 and 20. We use parametric LIF neuron with soft reset and retain the backpropagation of reset path in SNNs, where SpikingJelly [56] is used to implement the model. The loss weights λ_{pose}, λ_{shape}, λ_{trans}, λ_{3D} and λ_{2D} are set to be 10, 1, 50, 1 and 10 respectively. We train the two models for 20 and 25 epochs respectively with batch size being 8. The learning rate starts from 0.01 and is scheduled by CosineAnnealingLR, with maximum epoch number of 21 and 26. The models are trained on a single NVIDIA A100 GPU. For testing, 1 and 8-second event streams are used for $T = 8$ and $T = 64$ models respectively.

Evaluation metrics. Similar to previous works [98], [266], we report three different metrics, mean per joint position error (MPJPE), pelvis-aligned MPJPE (PEL-MPJPE) and Procrustes-Aligned MPJPE (PA-MPJPE). MPJPE is defined as

$$\text{MPJPE} = \frac{1}{24}\sum_{j=1}^{24} \|\mathbf{J}_{\text{3D}}^{j} - \hat{\mathbf{J}}_{\text{3D}}^{j}\|_2, \tag{5.19}$$

where $\hat{\mathbf{J}}_{\text{3D}}, \mathbf{J}_{\text{3D}} \in \mathbb{R}^{24\times 3}$ are the predicted and target 3D joints. Pelvis-aligned MPJPE (PEL-MPJPE) means MPJPE after root joint alignment between the predicted and target pose, which is defined as

$$\text{PEL-MPJPE} = \frac{1}{24}\sum_{j=1}^{24} \|(\mathbf{J}_{\text{3D}}^{j} - \mathbf{J}_{\text{root,3D}}) - (\hat{\mathbf{J}}_{\text{3D}}^{j} - \hat{\mathbf{J}}_{\text{root,3D}})\|_2, \tag{5.20}$$

where $\mathbf{J}_{\text{root,3D}}$ and $\hat{\mathbf{J}}_{\text{root,3D}}$ are the root joint positions. Procrustes-aligned MPJPE (PA-MPJPE) means MPJPE after aligning the predicted pose with the target by a rigid transformation (R, t), which is defined as

$$\text{PA-MPJPE} = \frac{1}{24}\sum_{j=1}^{24} \|\mathbf{J}_{\text{3D}}^{j} - (\hat{\mathbf{J}}_{\text{3D}}^{j} \cdot R + t)\|_2. \tag{5.21}$$

Note that (R, t) are calculated by minimizing the average distance between the target 3D joints $\mathbf{J}_{\text{3D}}$ and the transformed predicted joints $(\hat{\mathbf{J}}_{\text{3D}} \cdot R + t)$.

Method	ANNs/ SNNs	Input	Params	T=8 (1 sec)				
				FLOPs	Engy	MPJPE ↓	PEL-MPJPE ↓	PA-MPJPE ↓
VIBE [98]	ANNs	V	48.3M	43.4G	0.19	-	73.1	50.9
MPS [220]		V	39.6M	45.6G	0.20	-	68.0	48.2
EventCap(VIBE) [227]	ANNs	V+E	48.3M	185.0G	0.85	-	71.9	50.4
EventCap(MPS) [227]		V+E	39.6M	187.2G	0.86	-	66.6	47.8
EventHPE(VIBE) [266]		G+E	49.0M	49.0G	0.22	-	69.6	48.9
EventHPE(MPS) [266]		G+E	39.6M	49.3G	0.22	-	65.1	46.5
†EventHPE(GT) [266]		G+E	46.9M	-	-	71.8	55.0	43.9
ResNet-GRU [98]	ANNs	E	46.9M	43.6G	0.20	111.2	60.0	45.3
ResNet-TF [26]		E	41.3M	50.5G	0.23	108.5	59.9	**44.1**
Video-Swin [115]		E	48.9M	44.7G	0.20	124.1	66.5	49.0
SEW-ResNet-TF	Mix	E	47.0M	24.5G	0.097	110.8	58.9	44.2
ANN2SNN [171]	SNNs	E	46.9M	12.5G	0.011	140.3	74.1	55.8
SEW-ResNet [57]		E	**25.8M**	9.1G	0.008	116.8	62.5	48.3
MA-SNN [235]		E	30.2M	**7.5G**	**0.007**	115.2	61.6	47.6
SpikeFormer [262]		E	36.8M	13.2G	0.011	112.5	60.2	46.8
Ours	SNNs	E	47.7M	9.4G	0.008	**107.1**	**58.8**	**44.1**

Table 5.4: Quantitative results of human pose tracking on the MMHPSD test set with T being 8.

We also report FLOPs in terms of the number of MAC or AC and energy consumption (Engy) in term of joule J to show the cost-efficiency of SNNs.

Comparison with SOTA Methods. We compare our method with four prior works to highlight the competency of using event signals only for human pose tracking: VIBE [98], MPS [220], EventCap [227], and EventHPE [266], where VIBE [98] and MPS [220] are two most recent methods for video-based human pose estimation. In Tab. 5.4 and 5.5, we use V, G and E to represent the input data of gray-scale video, first gray-scale image and event streams respectively. VIBE and MPS are applied as the most recent video-based baselines with ResNet50 as the backbone. Note that both methods use weak camera model without global translation, so we will not report their MPJPE. To extract initial poses from the gray-scale video as required by EventCap, we make use of pre-trained VIBE and MPS methods for the extraction, labeled as EventCap(VIBE) and EventCap(MPS). Since the authors have not published their code, we re-implement EventCap using PyTorch LBFGS optimizer and PyTorch3D differential render, following [266]. Besides, EventCap

Method	ANNs/ SNNs	Input	Params	T=64 (8 secs)				
				FLOPs	Engy	MPJPE ↓	PEL-MPJPE ↓	PA-MPJPE ↓
VIBE [98]	ANNs	V	48.3M	344.9G	1.58	-	75.4	53.6
MPS [220]		V	39.6M	348.3G	1.60	-	69.2	50.1
EventCap(VIBE) [227]	ANNs	V+E	48.3M	1477.7G	6.79	-	74.1	52.9
EventCap(MPS) [227]		V+E	39.6M	1481.1G	6.81	-	68.1	49.5
EventHPE(VIBE) [266]		G+E	49.0M	354.0G	1.62	-	71.6	50.2
EventHPE(MPS) [266]		G+E	39.6M	354.2G	1.63	-	66.8	48.1
†EventHPE(GT) [266]		G+E	46.9M	-	-	74.5	58.1	45.3
ResNet-GRU [98]	ANNs	E	46.9M	348.6G	1.60	115.0	64.2	49.5
ResNet-TF [26]		E	41.3M	403.8G	1.85	114.2	66.0	50.1
Video-Swin [115]		E	48.9M	359.6G	1.65	130.9	72.5	53.1
SEW-ResNet-TF	Mix	E	47.0M	199.7G	0.79	113.2	65.3	49.3
ANN2SNN [171]	SNNs	E	46.9M	98.8G	0.089	148.2	81.1	60.9
SEW-ResNet [57]		E	**25.8M**	56.7G	**0.051**	122.8	66.3	52.3
MA-SNN [235]		E	30.2M	**55.3G**	0.055	120.1	64.8	48.9
SpikeFormer [262]		E	36.8M	96.3G	0.086	118.1	64.1	48.4
Ours	SNNs	E	47.7M	63.4G	0.058	**111.8**	**61.7**	**45.6**

Table 5.5: Quantitative results of human pose tracking on the MMHPSD test set with T being 64.

is an iterative optimization approach, which typically requires much more FLOPs than end-to-end methods as is indicated in Tab. 5.4 and 5.5. For EventHPE, we also use VIBE and MPS for the starting pose extraction, denoted as EventHPE(VIBE) and EventHPE(MPS). We also report the results of EventHPE with ground-truth starting pose known, denoted as EventHPE(GT). This method is considered the upper bound as it is assumed to have perfect information of the starting pose in the first frame, without any pose errors induced by VIBE or MPS.

As shown in Tab. 5.4 and 5.5, the most recent MPS outperforms VIBE by approximately 9mm in PEL-MPJPE and 5mm in PA-MPJPE for both $T = 8$ and $T = 64$. This trend is also evident when comparing EventCap(MPS) with EventCap(VIBE) or EventHPE(MPS) with EventHPE(VIBE), indicating that the performance of the two prior works [227], [266] is significantly impacted by the accuracy of initial poses provided by the pre-trained video-based methods. When the initial poses are inaccurate, they might fall into a local minimum and only improve the initial states by up to 3mm in PEL-MPJPE and 2mm in

PA-MPJPE. In constrast, our end-to-end approach directly uses event streams, which are better to capture the motion dynamics than images. Consequently, our SNNs model achieves the best performance with the smallest gap to the upper bound EventHPE(GT) while using only about 6% of FLOPs required by the optimization-based EventCap and 20% of FLOPs needed by the EventHPE. This is further illustrated in Fig. 5.11, where the inaccurate initial or starting poses given by MPS lead to sub-optimal pose tracking outcomes compared to ours. We also provide a supplementary video[4] for better illustration of the results.

Comparison with ANNs models. To further illustrate the advantages of SNNs over ANNs in event-based human pose tracking, we compare our model with three popular ANNs models: ResNet with GRU used in [98], [266] (ResNet-GRU), ResNet with standard transformer [200] used in DETR [26] (ResNet-TF) and Video Swin Transformer proposed in [115] (Video-Swin). For fair comparisons, we select the architecture with about 45M parameters for all the models. The settings for training these ANNs models mostly follow those of our approach, except the learning rate, which starts from 0.0001 and is scheduled by StepLR with a 0.1 decay after 15 and 20 epochs for both T=8 and T=64, respectively. This is because ANNs models do not converge well using a higher learning rate, such as 0.001.

For the models of $T = 8$ in Tab. 5.4 and 5.5, our approach achieves slightly lower pose errors than ResNet-GRU and Video-Swin, while presenting competitive performance with ResNet-TF, which also achieves 44.1mm in PA-MPJPE. In the case of $T = 64$, where longer temporal dependencies are necessary for perception, the performance decline of the three ANNs models is noticeably larger than that of our SNNs model, with over 4.1mm vs. 1.5mm drop in PA-MPJPE. Furthermore, our model requires only 9.4G and 63.4G FLOPs for $T = 8$ and $T = 64$, respectively, which is less than 20% of the FLOPs needed by the three ANNs models. These results demonstrate the superiority of our SNNs approach in efficiently encoding long-term temporal dependencies within event streams, primarily due to the fundamentally different working

[4]Link of supplementary video.

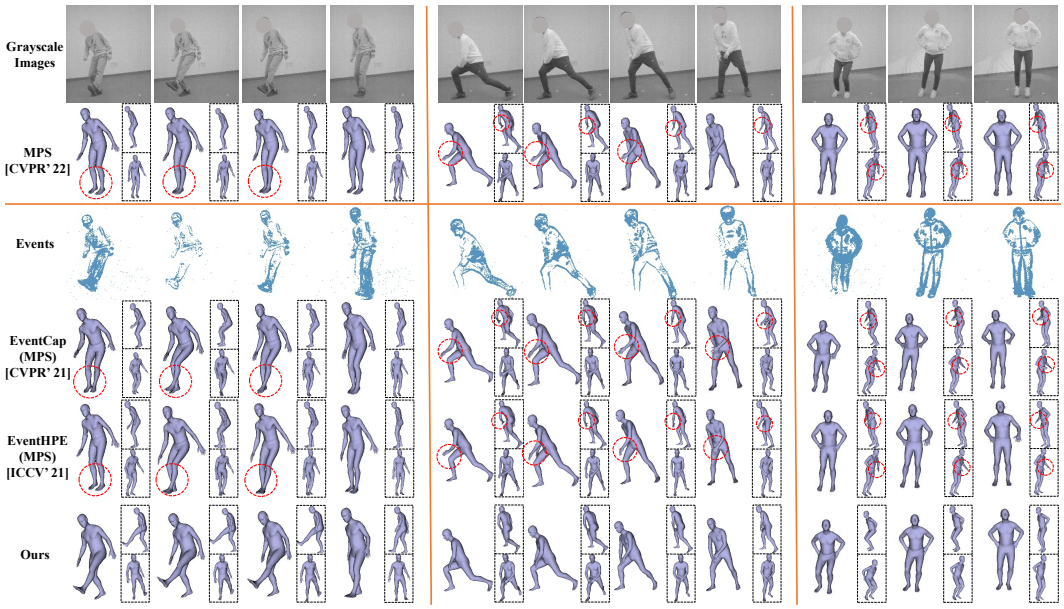


Figure 5.11: Qualitative results of ours compared with state-of-the-art methods.

mechanisms of spiking neurons compared to conventional artificial neurons.

Comparison with SNNs models. In Tab. 5.4 and 5.5, we compare our approach with five recently proposed SNNs models to highlight the superiority of our Spiking Spatiotemporal Transformer for human pose tracking. SEW-ResNet-TF acts as a baseline of mixed architecture that employs SEW-ResNet as the SNNs backbone followed by an ANN-based standard Transformer for pose tracking. The model architecture settings are similar to our approach. ANN2SNN refers to the conversion of the ANNs model of ResNet-GRU to the SNNs model using [171]. SEW-ResNet [57] is the backbone used in our approach without Spiking Spatiotemporal Transformer. MA-SNN [235] represents multi-dimensional attention SNNs where SEW-ResNet50 is used for a fair comparison. SpikeFormer [262] indicates an SNNs-based vision transformer (ViT) [50], where dot-product is directly adopted in the self-attention module.

Compared to the mixed model SEW-ResNet-TF, our approach exhibits slightly lower pose errors while requiring less than 50% of FLOPs. As for the entirely SNNs-based models, although ANN2SNN has shown its excellent performance in the image classification task, it falls short in the regression

task of pose tracking, producing much higher pose errors than other directly trained SNNs. This is primarily due to the quantization errors introduced during the conversion process. When compared to SEW-ResNet, our approach yields much lower pose errors – 48.3 vs. 44.1mm in PA-MPJPE for $T = 8$ – and the performance gap widens for $T = 64$, at 52.3 vs. 45.6mm. This demonstrates the importance of bi-directional temporal information provided by the proposed Spiking Spatiotemporal Transformer. In terms of MA-SNN, although it requires fewer FLOPs than our approach due to its lower spiking rate of 16.4% vs. ours of 22.6%, its performance is still inferior. Additionally, our approach presents moderately lower pose errors and fewer FLOPs than SpikeFormer, which is attributed to the proposed normalized Hamming similarity in the spiking attention module, as opposed to the ill-posed dot-product between spike tensors.

5.4.2 Empirical Results on SynEventHPD Dataset

Although this dataset covers a variety of motions, as illustrated in Fig. 5.7, the potential domain gap between synthetic and real events data remains an open question. In this section, we aim to demonstrate the value of our SynEventHPD dataset. We select five models from Tab. 5.4 and 5.5 including one ANNs model (ResNet-GRU [98]), one mixed model (SEW-ResNet-TF) and three SNNs models (SEW-ResNet [57], MA-SNN [235] and Ours). All models are evaluated on the real MMHPSD test set, but trained using either the real MMHPSD train set, the synthetic SynEventHPD dataset or a combination of both synthetic dataset and the real train set.

The quantitative results are displayed in Tab. 5.6. It is evident that, compared to models trained using the real MMHPSD train set, the pose errors are generally higher for models trained on the synthetic SynEventHPD dataset. This is largely due to the domain gap between the synthetic and real events, which results in inferior performance when training only on synthetic data and then evaluating on real data. However, after combining both real and synthetic datasets for training, all the models in Tab. 5.6 achieve improved performance compared to using either the real MMHPSD train set or the synthetic Syn-

Model	ANN/ SNN	Training Set	T=8 (1 sec)		
			MPJPE ↓	PEL-MPJPE ↓	PA-MPJPE ↓
ResNet-GRU [98]	ANN	Syn	113.6	62.2	47.5
		Real	111.2	60.0	45.3
		Syn&Real	105.4 (5.8)	58.9 (1.1)	44.6 (0.7)
SEW-ResNet-TF	Mix	Syn	114.1	60.6	45.5
		Real	110.8	58.9	44.2
		Syn&Real	104.2 (6.6)	58.4 (0.5)	43.5 (0.7)
SEW-ResNet [57]	SNN	Syn	120.3	63.6	49.1
		Real	116.8	62.5	48.3
		Syn&Real	113.1 (3.7)	61.7 (0.8)	47.8 (0.5)
MA-SNN [235]		Syn	119.0	63.1	48.8
		Real	115.2	61.6	47.6
		Syn&Real	112.5 (2.7)	60.7 (0.9)	46.9 (0.7)
Ours		Syn	110.7	59.4	45.0
		Real	107.1	58.8	44.1
		Syn&Real	103.1 (4.0)	58.4 (0.4)	43.8 (0.3)

Table 5.6: Quantitative results on the real MMHPSD test set with models trained on real/synthetic datasets.

EventHPD dataset alone. This highlights the effectiveness of the proposed SynEventHPD dataset. We also present qualitative results in Fig. 5.12, illustrating the performance of our model trained solely on the synthetic SynEventHPD dataset and applied to unseen scenarios. The left two examples show predictions on synthetic events generated from webcam videos, while the right example displays test results on real data. Despite being trained on the synthetic dataset, our model still demonstrates its generalization ability and applicability.

5.4.3 Empirical Results on DHP19 Dataset

DHP19 dataset [23] only provides 2-view events stream and joint positions without available gray-scale images and SMPL pose and shape parameters. As a result, we follow the settings in [23] where event frames are the input and the 2D joint heatmaps are the output. Then, using the predicted 2-view 2D joint positions, 3D pose can be reconstructed. We report the joint errors in Tab. 5.7, where DHP19 [23] uses CNNs to regress the joint heatmaps, SEW-ResNet [57] uses SNNs and Ours uses SNNs with our proposed spiking spatiotemporal transformer. Unlike SMPL pose, the 3D pose in this dataset is

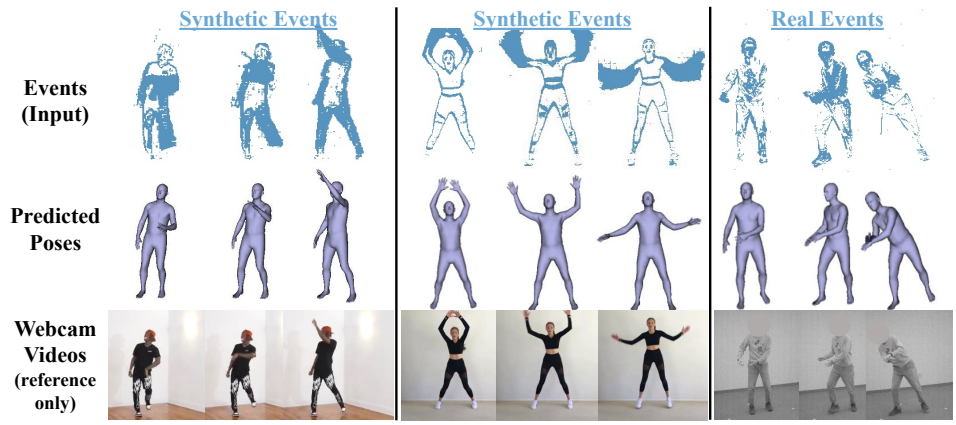


Figure 5.12: Generalization ability of our model, trained solely on the synthetic SynEventHPD dataset and applied to unseen scenarios.

Model	ANN/ SNN	T=8 (1 sec)			
		FLOPs	MPJPE ↓	PEL-MPJPE ↓	PA-MPJPE ↓
DHP19 [23]	ANN	42.4G	80.1	131.7	74.6
SEW-ResNet [57]	SNN	8.7G	75.7	118.1	70.9
Ours	SNN	10.1G	70.7	110.1	68.6

Table 5.7: Quantitative results on DHP19 dataset.

unconstrained, resulting in the higher joint errors after aligning root joint. The quantitative results show that our approach and SEW-ResNet exceed DHP19 by a large margin while requiring less than 25% FLOPs, which could attribute the properties of temporal dependencies in SNNs. When comparing to SEW-ResNet, our approach still gives much lower joint errors, demonstrating the effectiveness of our proposed spiking spatiotemporal transformer for better bi-directional temporal information fusion.

5.4.4 Ablation Study

In this section, we perform ablation studies to assess several crucial components in our approach. The quantitative results can be found in the corresponding sub-figures of Fig. 5.13.

(i) Score function in Spiking Spatiotemporal Transformer: We compare the proposed normalized Hamming similarity between spike vectors to scaled dot-production similarity, normalized Euclidean similarity and nor-

malized Manhattan similarity as detailed below:

$$\text{Nomalized Hamming similarity} \quad 1 - \frac{1}{C_k} \sum_{c=1}^{C_k} \mathbb{1}(\mathbf{s}_{ic}^q \neq \mathbf{s}_{jc}^k),$$

$$\text{Scaled dot-product similarity} \quad \frac{1}{\sqrt{C_k}} \sum_{c=1}^{C_k} \mathbf{s}_{ic}^q \cdot \mathbf{s}_{jc}^k,$$

$$\text{Normalized Euclidean similarity} \quad 1 - \frac{1}{C_k} \sum_{c=1}^{C_k} (\mathbf{s}_{ic}^q - \mathbf{s}_{jc}^k)^2,$$

$$\text{Normalized Manhattan similarity} \quad 1 - \frac{1}{C_k} \sum_{c=1}^{C_k} |\mathbf{s}_{ic}^q - \mathbf{s}_{jc}^k|.$$

The quantitative results of PEL-MPJPE depicted in Fig. 5.13 (i) reveal that our approach outperforms the other three commonly used similarity functions by over 3mm, showcasing the effectiveness of the normalized Hamming similarity as the score function for spike vectors.

(ii) Channel C of input voxel: We compare channel sizes of 1, 2, 4, 6 and 8 in terms of PEL-MPJPE in Fig. 5.13 (ii). The results show that $C = 4$ yields lower joint errors compared to sizes of 1 and 2, while nearly the same errors as sizes 6 and 8. Therefore, $C = 4$ is empirically determined to be the appropriate choice for the channel size.

(iii) # of attention layers: We compare our model with 0, 1, 2, 4 and 6 layers in Spiking Spatiotemporal Transformer in Fig. 5.13 (iii). The improvement of PEL-MPJPE is noticeable when using 1 or 2 layers of attention in our spiking transformer. However, this improvement is minimal for 4 and 6 layers, accompanied by a dramatic increase in the number of parameters from 47.7M to 87.4M.

5.4.5 Discussions

Attention scores maps are shown in Fig. 5.14. For better visualization, we transform the attention score matrix from $THW \times THW$ to $T \times T$, where the attention weights of spatial positions at each time step are summed together. The two examples illustrate that our attention mechanism allows the query at $t = 1$ to focus predominantly on features originating from subsequent time steps, thereby providing a more accurate and efficient prediction of body part

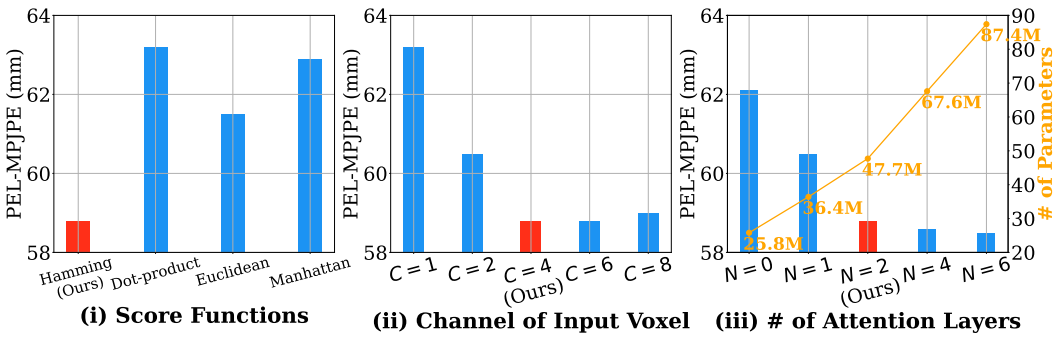


Figure 5.13: Ablation studies of three components in our proposed Spatiotemporal Spiking Transformer.

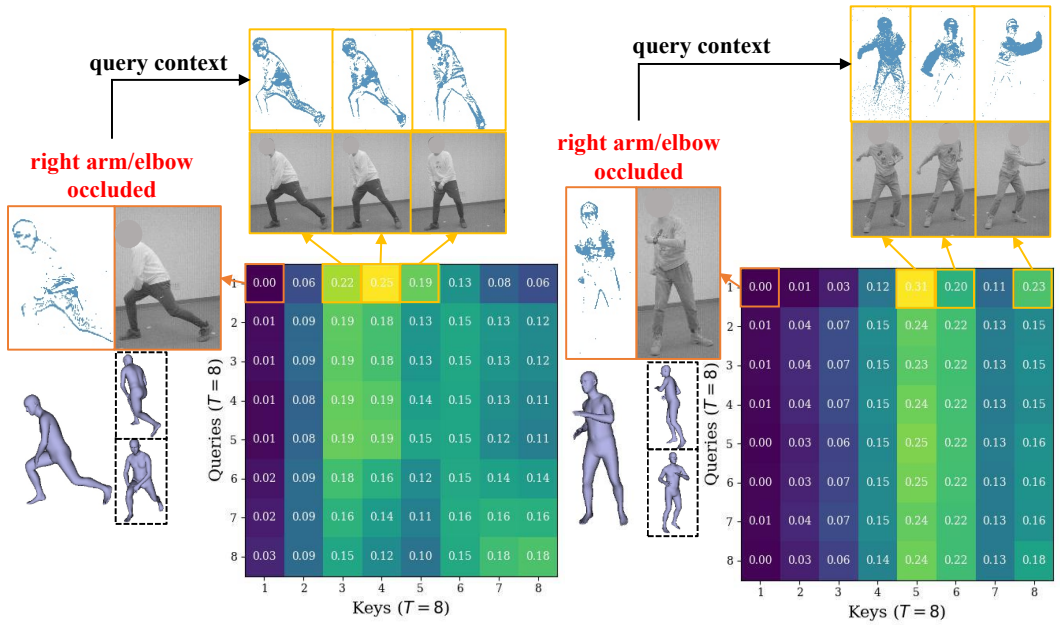


Figure 5.14: Visualization of attention score maps.

positions even when they are obscured. The success of these examples can be attributed to its ability to globally adapt to the temporal dependencies present in the input event stream. By emphasizing the relevant features from temporal context, our method can effectively compensate for the lack of information due to occlusion in the initial stages. This results in more accurate and robust pose tracking through time from events only.

Failure cases are displayed in Fig. 5.15, where the pose are not accurately estimated from the events. These cases were primarily attributed to the presence of body part occlusion and the absence of temporal context. The impact

of occlusion can be significant, as it hinders the model's ability to detect and analyze essential features required for pose estimation. Moreover, unlike the examples in Fig. 5.14, the lack of temporal context further compounds this issue, as the model cannot effectively leverage information from previous or subsequent frames to compensate for missing or obscured data. Recognizing and addressing these failure cases is crucial for improving the robustness and reliability of our event-based pose tracking method.

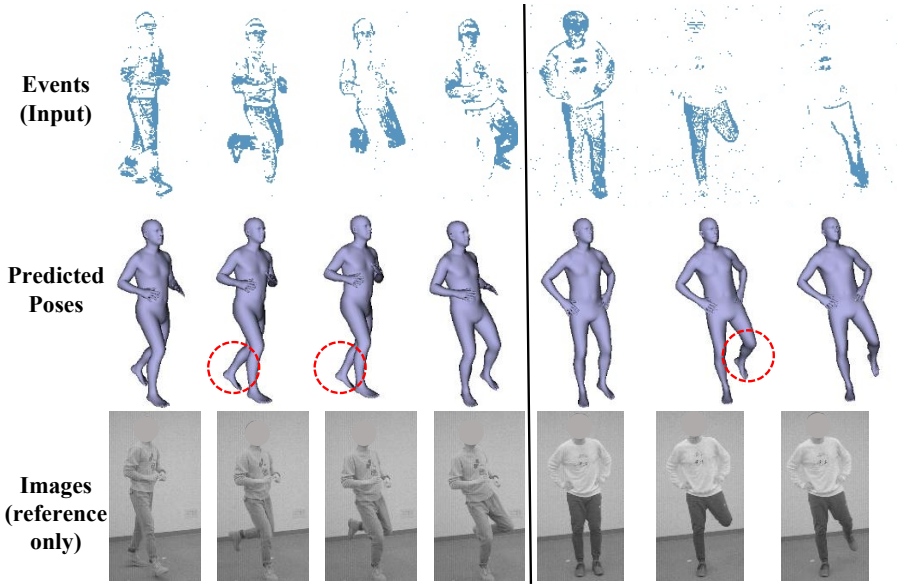


Figure 5.15: Failure cases.

5.5 Conclusion

We present in this chapter a dedicated end-to-end SNNs approach for event-based pose tracking, where events are the only source of input, thus removing the need of additional RGB or gray-scale images. Our approach is based entirely upon SNNs, with the proposed Spiking Spatiotemporal Transformer demonstrating its effectiveness for bi-directional temporal feature compensation. A large-scale synthetic dataset is also constructed, featuring a broad and diverse set of annotated 3D human motions, as well as longer hours of event stream data. Empirical experiments demonstrate the superiority of our approach in both efficacy and efficiency measures.

Chapter 6

Human Pose and Shape Estimation from Single Polarization Images

6.1 Introduction

A critical computer vision problem is to predict 3D human poses, *i.e.*, 3D body joint locations, from single images. In recent years, rapid progress is made in 3D human pose prediction from *RGB* images [55], [70], [90], [144], [146], [150], [209], [212], [249], [252]. Moreover, fueled by the development in parametric human shape modelling, such as SCAPE [7] and SMPL [116], it becomes feasible to estimate human body shapes from a single RGB image, as is evidenced by a number of end-to-end deep learning methods [10], [19], [48], [91], [92], [97], [101], [117], [156], [253], [257], [264], [271]. On the other hand, the problems of 3D human pose and shape estimation from single RGB images are still far from being solved. This is mainly due to the inherent lack of 3D cues in an RGB image. Furthermore, as the human shape models (e.g. SMPL) are usually learned from large sets of scanned human naked bodies, they are often lacking in clothing details.

The above observation inspires us to investigate a new imaging modality, polarization images, in this paper. That is, we consider the problem of estimating human pose and shape from a single polarization image. Polarization camera is built on a basic physics principle: a light ray reflected from an object is usually polarized. The polarized signal thus carries sufficient geometric

cues of object surface details to reliably infer its surface normal [9], [231]. It is worth mentioning the biological fact that light polarization could be directly perceived by some species of bees, ants, and shrimps for purposes such as 3D navigation [42], [218]. Motivated by the physical and biological facts, we propose a dedicated two-stage approach for human pose and shape estimation by integrating the geometric cues from the input polarization images.

As shown in Fig. 6.1, our approach, also called HumanSfP, contains two main stages. Stage one, Polar2Normal, concentrates on predicting accurate surface normal maps from single polarization images by exploiting the associated physics laws as priors. It is then fed into stage two, Polar2Shape, to reconstruct a clothed human shape guided by the obtained surface normal and an initial SMPL naked shape.

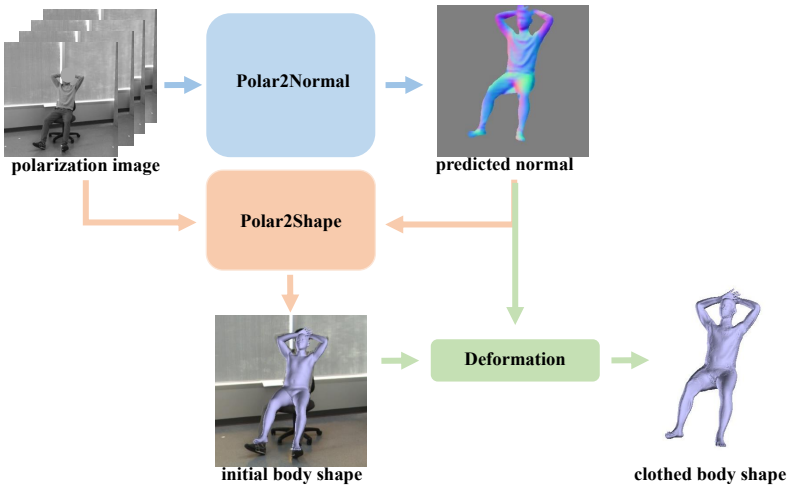


Figure 6.1: An overview of our HumanSfP approach that consists of two main stages: Polar2Normal and Polar2Shape.

Different from previous efforts in estimating detailed depth map [192], [264], our stage one focuses on surface normal estimation from a polarization image. By explicitly integrating the underlying physical principles, it gives rise to more reliable estimation. To achieve this goal, two main challenges need to be addressed, namely π-ambiguity of the azimuth angle and the possible environmental noise in practical applications. Based on the physical laws of polarization, two ambiguous normal maps $\mathbf{m}_1$ and $\mathbf{m}_2$ (Sec. 6.3.1) can be di-

rectly calculated from a polarization image. It is often reasonable to assume light reflected by human clothes is dominated by diffused reflection. Then, different from [9], we consider a two-branch strategy: one branch is employed to categorize each pixel into three categories: the two ambiguous normal maps and background; the second branch is to infer a coarse surface normal map. A fused normal map is produced by incorporating the classification results as well as the two ambiguous normal maps. Finally, as is revealed by [269], the fused normal map might still be noisy, owing to the environmental noise and the digital quantization of the polarization camera. Instead, we work with the normal residual, calculated as the difference between the coarse and the fused normal maps, to refine the coarse normal map and produce our final normal estimation.

Based on the raw polarization image and final surface normal output of stage one, stage two concerns the estimation of 3D human pose and the reconstruction of its clothed shape. It starts from estimating an initial parametric shape, *i.e.*, SMPL shape, as 3D pose, which is then deformed by leveraging the geometric details from the surface normal to reconstruct the final clothed human shape. Different from previous works [91], [92], [269], geodesic distance is employed in SMPL shape estimation, since the pose representation of SMPL is naturally in the product space of $SO(3)$, a classical example of Lie group. Empirical evidence shows that our two-stage pipeline can faithfully infer detailed surface normals and accurately estimate human poses and clothed body shapes.

To summarize, there are three main contributions in our work:

- A new problem, namely human pose and shape estimation from single polarization images, is proposed. A dedicated deep learning approach, HumanSfP, is proposed, where the detail-preserving surface normal maps are obtained following the physical laws of light polarization, and are shown to estimate more accurate pose and body shape.

- In tackling this new problem, a dedicated Polarization Human Shape and Pose Dataset, PHSPD, has been created. It now consists of ∼527K

frames and their corresponding pose and shape annotations. Overall there are 21 different subjects performing 31 unique actions, and ∼9.5 hours of videos are recorded in total. [1].

- Empirical evaluations on a synthetic dataset, SURREAL dataset, as well as our real-world dataset, PHSPD dataset, demonstrate the effectiveness and applicability of our approach. Our work showcases that, for estimating 3D human poses and shapes, a 2D polarization camera could be a viable alternative to a conventional RGB camera.

6.2 Preliminary Backgrounds

6.2.1 Polarization Image Formation

The light reflected from an object's surface mainly includes three components [41], the polarized specular reflection, the polarized diffuse reflection, and the unpolarized diffuse reflection. A polarization camera is equipped with an array of linear polarizers mounted right on top of its CMOS imager, in place of the RGB Bayer filters. During the imaging process, a pixel's intensity typically varies sinusoidally with the angle of the polarizer [231]. In this work, we assume that the light reflected from human clothes is dominated by polarized diffuse reflection and unpolarized diffuse reflection. Then, for a specific polarizer angle ϕ_{pol}, the illumination intensity at a pixel with dominant diffuse reflection is

$$\mathrm{I}(\phi_{\text{pol}}) = \frac{\mathrm{I}_{\max} + \mathrm{I}_{\min}}{2} + \frac{\mathrm{I}_{\max} - \mathrm{I}_{\min}}{2} \cos\big(2(\phi_{\text{pol}} - \varphi)\big). \tag{6.1}$$

Here φ is azimuth angle of the surface normal, $\mathrm{I}_{\max}$ and $\mathrm{I}_{\min}$ are the upper and lower bounds of the illumination intensity. $\mathrm{I}_{\max}$ and $\mathrm{I}_{\min}$ are mainly determined by the unpolarized diffuse reflection, and the sinusoidal variation is mainly determined by the polarized diffuse reflection. If the intensity images $\mathrm{I}(\phi_{\text{pol}})$ under three or more different polarizer angles can be obtained, such as $\mathrm{I}(0°)$, $\mathrm{I}(45°)$ and $\mathrm{I}(90°)$, φ can be solved in closed form. Note that there

[1] The dataset and our code are publicly available at https://github.com/JimmyZou/PolarHumanPoseShape

is π-ambiguity in the azimuth angle φ in Eq. (6.1), which means that φ and $\pi + \varphi$ will result in the same illumination intensity of the pixel. As for the zenith angle θ, it is related to the degree of polarization ρ,

$$\rho = \frac{\mathrm{I}_{\max} - \mathrm{I}_{\min}}{\mathrm{I}_{\max} + \mathrm{I}_{\min}}. \tag{6.2}$$

According to [8], when diffuse reflection dominates, the degree of polarization ρ becomes a function of the zenith angle θ and the refractive index n,

$$\rho = \frac{(n - \frac{1}{n})^2 \sin^2\theta}{2 + 2n^2 - (n + \frac{1}{n})^2 \sin^2\theta + 4\cos\theta\sqrt{n^2 - \sin^2\theta}}. \tag{6.3}$$

The solution of θ in Eq. (6.3) is thus a close-form expression of n and ρ.

The refractive index of human clothing typically varies between 1.3 and 1.5 [134]. In most instances, polarized diffuse reflection is the predominant component of light reflected from clothing. Beyond clothing, other human surfaces captured by polarization cameras include skin, such as faces and limbs exposed when wearing t-shirts or shorts. The refractive index of human skin is similar, ranging from 1.35 to 1.55 [134]. However, there are instances, particularly when skincare products are applied to the face, where polarized diffuse reflection becomes a major component. This variation can slightly impact our approach, which primarily assumes that polarized diffuse reflection dominates. It is important to note that our dataset contains only a limited number of such cases.

6.2.2 Special Orthogonal Group

Mathematically, a Lie group [139] is a group as well as a smooth manifold. 3D rotation transformations, also known as the Special Orthogonal group $SO(3)$, is exactly a Lie group and could be characterized by

$$SO(3) = \{R \in \mathbb{R}^{3\times3} | R^\top R = I, \det(R) = +1\}. \tag{6.4}$$

The tangent space of Lie group $SO(3)$ at identity I_3 is referred to as its Lie algebra $\mathfrak{so}(3)$. An element of $\mathfrak{so}(3)$ is a 3×3 skew-symmetric matrix $\hat{W}$ defined as

$$\hat{W} = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix}. \tag{6.5}$$

Essentially, $\mathfrak{so}(3)$ spans a 3-dimensional vector space, denoted by $\mathbf{w} = (w_1, w_2, w_3)^\top$. The mapping from a Lie algebra vector $\hat{W} \in \mathfrak{so}(3)$ to a point in the manifold $R \in SO(3)$ is formulated as an exponential map $\exp : \mathfrak{so}(3) \to SO(3)$ as

$$R = \exp(\hat{W}) = \mathrm{I} + \frac{\sin(\|\mathbf{w}\|)}{\|\mathbf{w}\|}\hat{W} + \frac{1 - \cos(\|\mathbf{w}\|)}{\|\mathbf{w}\|}\hat{W}^2, \tag{6.6}$$

where $\|\cdot\|$ denotes the vector norm. The geodesic distance of two points in the manifold, $R_1, R_2 \in SO(3)$, is defined as the angular difference between the two rotations, which is

$$D(R_1, R_2) = \left|\cos^{-1}\left(\frac{\mathrm{Tr}(R_1^\top R_2) - 1}{2}\right)\right|. \tag{6.7}$$

6.3 Our HumanSfP Approach

There are two main stages in our approach. 1) Stage one shown in Fig. 6.2 is our Polar2Normal pipeline for surface normal estimation from a polarization image. After inferring two ambiguous normal maps, ($\mathbf{m}_1$, $\mathbf{m}_2$), as physical priors from the polarization image (see Sec. 6.3.1 for details), a two-branch strategy is adopted: one branch classifies each image pixel as belonging to either of the two normal maps or the background, thus obtaining the fused normal $\mathbf{m}_3$; a second branch regresses the coarse normal map $\mathbf{m}_4$ as an intermediate result. They are followed by the final step, which focuses on the residual refinement of coarse normal map to integrate the fused and the coarse normal maps as well as the normal residual to regress the final surface normal. Note that modules in the gray dash-line box are specifically designed for polarization images, to leverage the physical prior knowledge that reflected light from an object is polarized; these modules are unfit in dealing with RGB images. 2) Stage two shown in Fig. 6.3 is our Polar2Shape pipeline of clothed body shape reconstruction from a polarization image, accomplished in two steps. The first step focuses on estimating the parameters of SMPL model, a rough & naked shape model parameterized by Θ. The next step is to deform the initial SMPL shape according to the estimated surface normal in Sec.6.3.1, to reconstruct the refined 3D human shape with clothing details.

6.3.1 Polar2Normal: Surface Normal Estimation

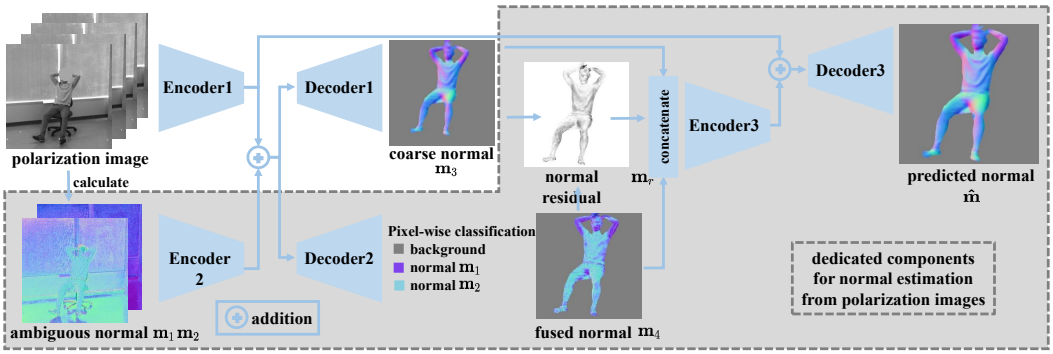


Figure 6.2: Our Polar2Normal pipeline for surface normal estimation from a polarization image.

The polarization image in our work consists of four channels, and each channel corresponds to one of the four polarizer angles: 0°, 45°, 90° and 135°. Taking into account the π-ambiguity of φ, we have two possible solutions to the surface normal for each pixel, that form the physical priors, denoted as ambiguous normal maps $\mathbf{m}_1(\varphi, \theta)$ and $\mathbf{m}_2(\pi + \varphi, \theta)$. We propose a two-step architecture to estimate the surface normal from the polarization image. The details are presented in Fig. 6.2.

In the first step, two encoders are used to extract visual features from the polarization image and the two ambiguous normal maps separately, which are then followed two decoders. One decoder is to capture a coarse surface normal of the human body denoted by $\mathbf{m}_3$, where Huber loss is employed to train the network, defined as

$$H(x, \alpha) = \begin{cases} 0.5\, x^2, & |x| < \alpha \\ \alpha\,(|x| - 0.5\,\alpha), & \text{otherwise.} \end{cases} \tag{6.8}$$

The loss of the coarse surface normal becomes

$$\mathcal{L}_{\text{coarse}} = \sum_{i,j} H\big(1 - \langle \mathbf{m}_{3[i,j]}, \mathbf{m}_{[i,j]} \rangle, \alpha\big), \tag{6.9}$$

where $\mathbf{m}$ is the target normal map with (i, j) being the pixel coordinate, $\langle \mathbf{m}_{3[i,j]}, \mathbf{m}_{[i,j]} \rangle$ is the cosine similarity between the two normal vectors, and α controls the trade-off between the squared and the absolute losses. Now,

the second decoder is to classify each pixel into three categories: background, ambiguous normal $\mathbf{m}_1$, and ambiguous normal $\mathbf{m}_2$, with the corresponding pixel-wise probabilities p_0, p_1, and p_2, respectively. The fused normal is thus obtained by

$$\mathbf{m}_4 = (1 - p_0) \cdot \frac{p_1\mathbf{m}_1 + p_2\mathbf{m}_2}{\|p_1\mathbf{m}_1 + p_2\mathbf{m}_2\|_2}, \tag{6.10}$$

where $1 - p_0$ acts as a soft mask for the foreground human body. The classification loss is measured by the cross entropy between the predicted pixel-wise category and the target category,

$$\mathcal{L}_{\text{category}} = \sum_{i,j} \sum_{c} y_{c[i,j]} \log p_{c[i,j]} + (1 - y_{c[i,j]}) \log(1 - p_{c[i,j]}). \tag{6.11}$$

Here c indexes among the three categories. $y_{c[i,j]} \in \{0, 1\}$ is the multi-class label indicating which category the pixel $[i, j]$ belongs to. Note that the label $y_{c[i,j]}$ is created by discriminating whether the pixel is background or which ambiguous normal has higher cosine similarity with its target normal.

Next let us look at the second step. Different from our previous work [269] that directly regresses the normal map given the polarization image and fused normal, we propose a residual update scheme to produce a more detailed and accurate surface normal estimation, as displayed in Fig. 6.2. Due to the environmental noise and the digital quantization of the polarization image formation process, the fused normal map $\mathbf{m}_3$ is often noisy and non-smooth. Given the coarse normal map $\mathbf{m}_4$ and fused normal map $\mathbf{m}_3$, the normal residual $\mathbf{m}_r$ is evaluated by

$$\mathbf{m}_{r[i,j]} = 1 - \langle \mathbf{m}_{3[i,j]}, \mathbf{m}_{[i,j]} \rangle. \tag{6.12}$$

A denoising network is then trained to take both normal maps $\mathbf{m}_3, \mathbf{m}_4$ and the normal residual $\mathbf{m}_r$ as input, to produce a smoothed and detailed normal $\hat{\mathbf{m}}$. In the residual update scheme, the coarse normal $\mathbf{m}_3$ is replaced by the final predicted normal $\hat{\mathbf{m}}$ to produce the new normal residual $\mathbf{m}_r$, which is then employed to regress an updated normal map $\hat{\mathbf{m}}$. The loss for $\hat{\mathbf{m}}$ is defined by

$$\mathcal{L}_{\text{final}} = \sum_{i,j} \left| 1 - \langle \hat{\mathbf{m}}_{[i,j]}, \mathbf{m}_{[i,j]} \rangle \right|. \tag{6.13}$$

Here $|\cdot|$ denotes the absolute value.

Finally, our surface normal estimation model is learned by minimizing the following loss

$$\mathcal{L}_{\text{polar2normal}} = \mathcal{L}_{\text{coarse}} + \mathcal{L}_{\text{category}} + \mathcal{L}_{\text{final}}. \tag{6.14}$$

6.3.2 Polar2Shape: Human Shape Reconstruction

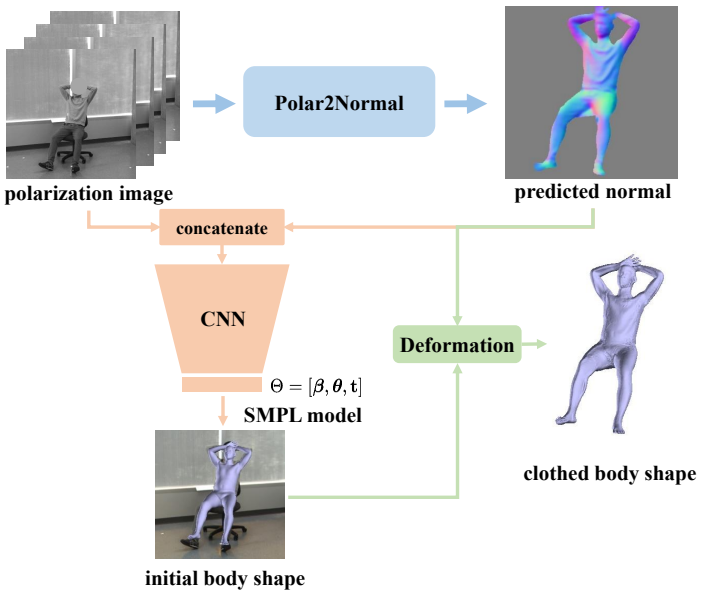


Figure 6.3: Our Polar2Shape pipeline of clothed body shape reconstruction from a polarization image, accomplished in two steps.

Stage two of our approach, also referred to as Polar2Shape, focuses on the reconstruction of clothed body shape from a polarization image, accomplished in two steps. In the first step, the naked initial body shape, represented by SMPL model [116], is estimated. The following step is to deform the initial human body shape with the estimated surface normal, and finally the clothed body shape is reconstructed. This process is shown in Fig. 6.3.

Initial Shape Estimation. The core of SMPL model [116] lies in a differentiable function $\mathcal{M}(\boldsymbol{\beta}, \boldsymbol{\theta}) \in \mathbb{R}^{6890\times 3}$ that outputs a triangular mesh with 6890 vertices from 82 parameters $[\boldsymbol{\beta}, \boldsymbol{\theta}]$. $\boldsymbol{\theta} \in \mathbb{R}^{72}$ are the pose parameters to characterize pose articulations in axis-angle representation, consisting of one global rotation of the body and the relative rotations of its 23 joints. Human

pose is therefore represented as $\boldsymbol{\theta} = (\boldsymbol{\theta}_j)_{j=1}^{24}$, where $\boldsymbol{\theta}_j \in \mathbb{R}^3$, and j denotes the index of relative rotation in axis-angle. The shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$ are the linear coefficients of a PCA shape space that mainly determine individual body features such height, weight and body proportions. The PCA shape space is learned from a large dataset of naked and minimal clothed human body scans. A specific SMPL shape is produced by first applying pose-dependent and shape-dependent deformations to the template pose, then using forward-kinematics to articulate the body to its current pose, and finally deforming the surface mesh by linear blend skinning. At the same time, the 3D joint positions, denoted by $\mathbf{J}_{3\mathrm{D}} \in \mathbb{R}^{24\times 3}$, are obtained by linear regression from the output mesh vertices. In addition to the SMPL model parameters, the global translation of the human body is also a necessary factor in aligning with the projection in the 2D image space, denoted by $\mathbf{t} \in \mathbb{R}^3$. This results in an 85-dimensional parameter space, $\Theta = [\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{t}]$, that are used by SMPL to dictate a specific intial human shape.

Under mild assumption, the axis-angle representation of human pose in SMPL, $(\boldsymbol{\theta}_j)_{j=1}^{24}$, has a bijective map to the corresponding 24-dim product space of $\mathfrak{so}(3)$ manifold. Existing efforts, such as [269], normally predict the pose in axis-angle representation, and measure the Euclidean distance between the predicted pose and target pose as $\sum_{j=1}^{24} \|\hat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j\|_2$. However, as the two poses corresponds to two points in the aforementioned curved space, their distance is better characterized by the geodesic distance, which is not necessarily the Euclidean distance. Thus we propose to represent the human pose as a set of rotation matrices $\{R_j\}_{j=1}^{24}$ in $SO(3)$ with $R_j = \exp(\boldsymbol{\theta}_j)$. The geodesic distance between the predicted pose and target pose becomes

$$\mathcal{L}_{\text{pose}} = \sum_{j=1}^{24} D(R_j, \hat{R}_j) = \sum_{j=1}^{24} \left| \cos^{-1}\left(\frac{\mathrm{Tr}(R_j^\top \hat{R}_j) - 1}{2}\right)\right|, \tag{6.15}$$

which is also referred as the pose estimation loss. Moreover, the losses for

SMPL shape, global translation and joint positions are defined as

$$\mathcal{L}_{\text{shape}} = \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\|_2^2, \tag{6.16}$$

$$\mathcal{L}_{\text{trans}} = \|\mathbf{t} - \hat{\mathbf{t}}\|_2^2, \tag{6.17}$$

$$\mathcal{L}_{\text{joints}} = \|\mathbf{J}_{\text{3D}} - \hat{\mathbf{J}}_{\text{3D}}\|_2^2. \tag{6.18}$$

At the moment, our initial shape estimation is learned by minimizing the following loss

$$\mathcal{L}_{\text{polar2shape}} = \mathcal{L}_{\text{pose}} + \lambda_{\text{shape}}\mathcal{L}_{\text{shape}} + \lambda_{\text{trans}}\mathcal{L}_{\text{trans}} + \lambda_{\text{joints}}\mathcal{L}_{\text{joints}}, \tag{6.19}$$

where λ_{shape}, λ_{trans} and λ_{joints} are the tuning parameters of the corresponding loss terms.

Shape Reconstruction. The initial human shape obtained by SMPL representation still lacks fine surface details. Therefore, the aim of this step is to refine the initial SMPL shape guided by our surface normal estimate, as follows. The SMPL body shape is rendered on the image plane to form an initial depth map. The technique of [140] is then engaged here to obtain an optimized depth map I_d from the predicted surface normal $\hat{\mathbf{m}}$ and the initial depth map $\hat{I}_d$ by minimizing the objective function,

$$E(I_d) = \lambda_n E_n(I_d) + \lambda_d E_d(I_d) + \lambda_s E_s(I_d), \tag{6.20}$$

which contains three energy terms. The first term, $E_n(I_d)$, ensures the predicted normal to be perpendicular to the tangents of the optimized depth surface,

$$E_n(I_d) = \sum_{x,y} T_{u[x,y]}\hat{\mathbf{m}}_{[x,y]} + T_{v[x,y]}\hat{\mathbf{m}}_{[x,y]}. \tag{6.21}$$

Here $[x, y]$ denotes a pixel coordinate. u and v represent the horizontal and vertical direction of the image plane, respectively. The tangents T_u and T_v are defined as

$$T_u = \left(\frac{1}{f_u}\left(\frac{\partial I_d}{\partial u}(u - p_u) + I_d\right), \frac{1}{f_v}\frac{\partial I_d}{\partial u}(v - p_v), \frac{\partial I_d}{\partial u}\right)^\top, \tag{6.22}$$

$$T_v = \left(\frac{1}{f_u}\frac{\partial I_d}{\partial u}(v - p_v), \frac{1}{f_v}\left(\frac{\partial I_d}{\partial v}(v - p_v) + I_d\right), \frac{\partial I_d}{\partial v}\right)^\top, \tag{6.23}$$

where f_u and f_v denote the focal length, p_u and p_v define the camera center coordinate, respectively. The second term, $E_d(I_d)$, encourages the optimized depth to be close to the initial depth,

$$E_d(I_d) = \sum_{x,y} \left[\Big((\frac{x - p_v}{f_v})^2 + (\frac{y - p_u}{f_u})^2 + 1 \Big) \big(I_{d[x,y]} - \hat{I}_{d[x,y]} \big) \right]^2. \tag{6.24}$$

The third and final term preserves smoothness of nearby pixels over the optimized depth map,

$$E_s(I_d) = \sum_{x,y} \sum_{[x',y'] \in \mathcal{N}(x,y)} \left\| I_{d[x,y]} - I_{d[x',y']} \right\|^2. \tag{6.25}$$

Our depth map is therefore obtained as a solution of the above mentioned linear least-squares system. Finally, our clothed body shape is produced by upsampling and deforming the SMPL mesh according to the Laplacian of the optimized depth map.

6.3.3 Our In-house PHSPD Dataset

To facilitate empirical evaluation of our approach in real-world scenarios, a home-grown dataset is curated, which is referred as Polarization Human Pose and Shape Dataset or PHSPD. The layouts of our multi-camera acquisition system for PHSPDv1 and v2 are shown in Fig. 6.4. In PHSPDv1, 7 cameras, comprising three RGB-Depth cameras and a polarization camera, are synchronized, and in PHSPDv2, we extend the number of cameras in our system to be 12, including five RGB-Depth cameras, a polarization camera and an event camera. In what follows, we start by presenting the early version, PHSPDv1, as well as the more recent addition, PHSPDv2.

PHSPDv1. It is the early version used in our preliminary work [269]. During PHSPDv1 data acquisition, a system of 4 soft-synchronized cameras is used, consisting of a polarization camera and three RGB-Depth cameras. 12 subjects are recruited in data collection, in which 9 are male and 3 are female. Each subject performs 3 different sets of actions (out of 18 distinct action types) for 4 times, plus an additional period of free-form motion at the end of each session. Thus for each subject, there are 13 short videos (around

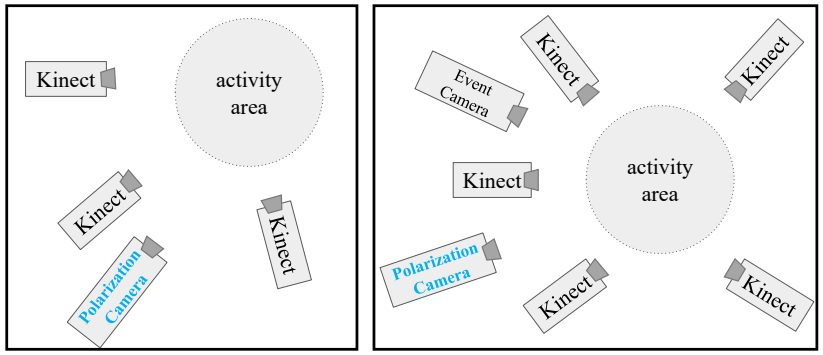


Figure 6.4: The layout of multi-camera system in our PHSPDv1 and PHSPDv2 datasets.

1,800 frames per video in 10-15 FPS). The total number of frames for each subject amounts to 22K. Overall, PHSPDv1 dataset consists of 287K frames. Each frame here contains a synchronized set of images: one polarization image, three RGB and Depth images. The examples are presented in Fig. 6.5 and all the types of actions are summarized in Tab. 6.1.

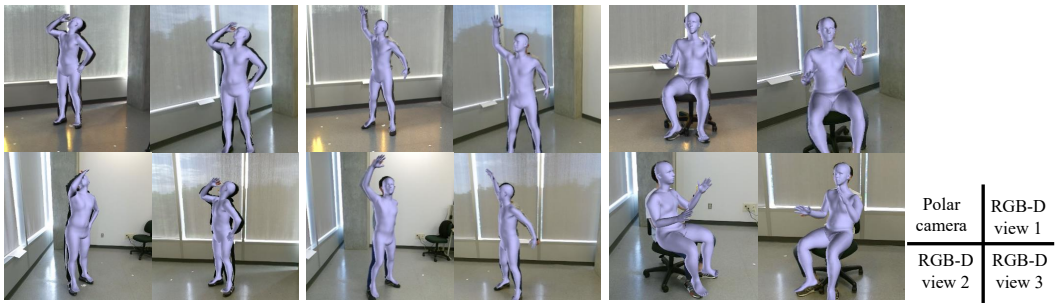


Figure 6.5: Exemplar multi-view figures with annotated shape and pose in PHSPDv1.

PHSPDv2. It contains the newly acquired data, where our multi-camera acquisition system is extended to 12 cameras of different modalities: one polarization camera, one event camera and five RGB-Depth cameras. 15 subjects are recruited for the data acquisition, where 11 are male and 4 are female. Each subject is required to perform 3 groups of actions (21 different actions in total) for 4 times, where each group includes actions of fast/medium/slow speed, respectively. Finally, 12 videos are collected for each subject and each video has around 1,300 frames in 15 FPS. In total, there are 180 videos, with

group	actions
1	warming-up, walking, running, jumping, drinking, lifting dumbbells
2	sitting, eating, driving, reading, phoning, waiting
3	presenting, boxing, posing, throwing, greeting, hugging, shaking hands

Table 6.1: Summary of action types performed by subjects in PHSPDv1.

group	speed	actions
1	medium	jumping, jogging, waving hands, kicking legs, walk
2	fast	boxing, javelin, fast running, shooting basketball, kicking football, playing tennis, playing badminton
3	slow	warming up elbow/wrist ankle/pectoral, lifting down-bell, squating down, drinking water

Table 6.2: Summary of action types performed by subjects in PHSPDv2.

each video lasting about 1.5 minutes. This amounts to 240k frames with each frame including one polarization image and five RGB & Depth images. The examples are presented in Fig. 6.6 and all the types of actions are summarized in Tab. 6.2.

To summarize, our PHSPD dataset contains 334 videos of 21 different subjects performing 31 types of actions. It totals approximately 527K frames, equivalent to about 9.5 hours of recorded footage. Each frame in the dataset contains a synchronized set of images including both a polarization image and RGB & Depth images.

Multi-camera Synchronization. The multi-camera system in our PHSPD is mostly soft synchronized. In PHSPDv1, each camera is connected with a desktop, where the desktop connected to the polarization camera is the master and the other three ones connected to three Kinects V2 are clients. The master desktop uses TCP-IP protocol to communicate with the other clients. After receiving a specific message, each client copies the most recent frame data captured by the Kinect into the desktop memory. At the same time, the master desktop sends a software trigger to the polarization camera to capture one frame into the camera buffer. For PHSPDv2, three latest RGB-Depth

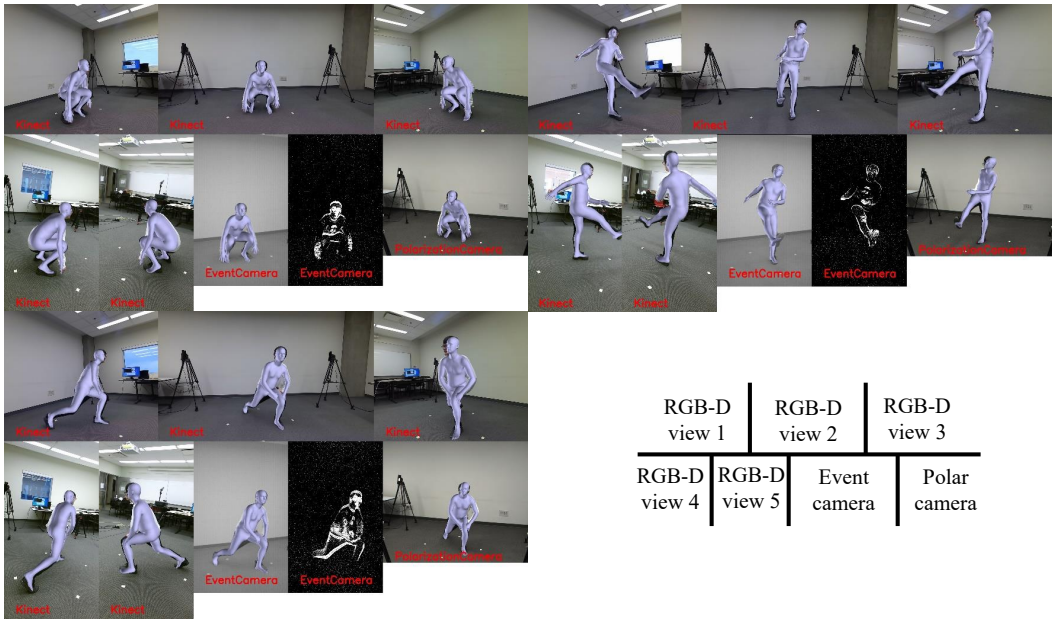


Figure 6.6: Exemplar multi-view figures with annotated shape and pose in PHSPDv2.

cameras, Azure Kinects [131], and two Kinects V2 are used. The three Azure Kinects are hard-synchronized and connected to one client desktop. Then, with three Azure Kinects, the other two Kinects are soft synchronized with the master desktop that is connected to a polarization camera and an event camera. Compared with expensive motion capture system used in [80], our multi-camera system does not require the subject to wear a number of sensors, which gives rise to a more natural appearance in the images.

Annotation. There are three main steps in the pipeline of our SMPL shape and pose annotation. 1) The first step is to obtain an initial 3D pose (joints position) from the multi-view RGB-Depth cameras. For each frame, the 2D joints of all the color images are detected by OpenPose [24] and the depth of each 2D joint is obtained by warping the depth image to the color image and finding the depth of its neighboring pixels. 2) The second step is to fit the SMPL male or female model to the initial pose via 3D SMPLify-x [154] and get the initial SMPL parameters. 3) The last step is to fine-tune the initial shape to fit the point-cloud collected from multi-view depth images using the L-BFGS [20] algorithm, where the average distance of shape vertex to

subject #	gender	raw #	annotated #	discarded #
1	female	22561	22241	320 (1.4%)
2	male	24325	24186	139 (0.5%)
3	male	23918	23470	448 (1.8%)
4	male	24242	23906	336 (1.4%)
5	male	24823	23430	1393 (5.6%)
6	male	24032	23523	509 (2.1%)
7	female	22598	22362	236 (1.0%)
8	male	23965	23459	506 (2.1%)
9	male	24712	24556	156 (0.6%)
10	female	24040	23581	459 (1.9%)
11	male	24303	23795	508 (2.1%)
12	male	24355	23603	752 (3.1%)
total	-	287874	282112	5762 (2.0%)

Table 6.3: Detail number of frames for each subject in PHSPDv1.

subject #	gender	raw #	annotated #	discarded #
1	male	15911	15911	0 (0.0%)
2	male	15803	15803	0 (0.0%)
3	male	16071	16071	0 (0.0%)
4	male	16168	16152	16 (0.01%)
5	male	16278	16262	16 (0.01%)
6	male	16715	16384	331 (2.0%)
7	female	16091	16091	0 (0.0%)
8	male	16257	15642	715 (4.4%)
9	male	15467	15461	6 (0.03%)
10	male	16655	16655	0 (0.0%)
11	male	16464	16443	21 (0.13%)
12	male	16186	16186	0 (0.0%)
13	female	16064	14562	1502 (9.4%)
14	female	15726	15166	560 (3.6%)
15	female	14193	14075	118 (0.8%)
total	-	240049	236764	3285 (1.4%)

Table 6.4: Detail number of frames for each subject in PHSPDv2.

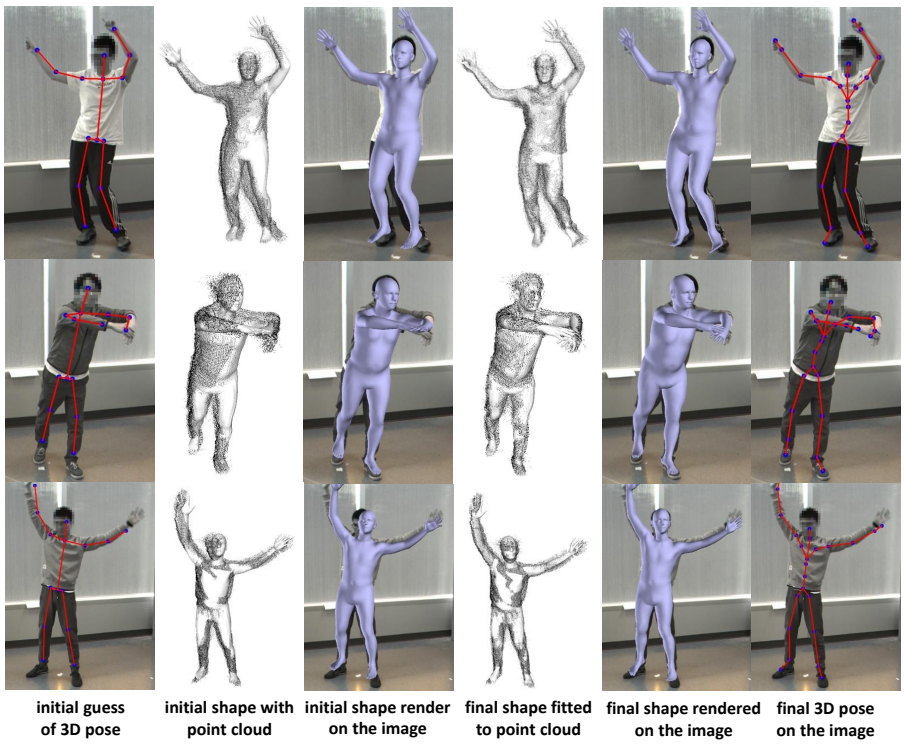


Figure 6.7: Exemplar figures to show that fine-tuning the initial SMPL shape to fit the point-clouds can give more accurate annotated shape and pose.

its nearest point in the point-cloud is minimized iteratively. Fig. 6.7 illustrates the effectiveness of the fine-tuning step to give more accurate shape and pose. Normally the initial pose is coarse because of the errors of depth or 2D joints detection. The iterative fine-tunning process can adjust the SMPL shape to fit the human body point-clouds better.

Empirically, we find that SMPL male model gives better annotations than female SMPL model for female subjects in light that our recruited female subjects have similar body proportion with SMPL male model. Thus our dataset adopts SMPL male model for all subjects. Some examples of multi-view annotated shape are displayed in Fig. 6.5 and 6.6. The details of annotation results for the PHSPDv1 and PHSPDv2 dataset are reported in Tab. 6.3 and 6.4.

Comparison with Existing Datasets. Our PHSPD dataset is compared side-by-side with six widely-used human pose and shape datasets in Tab. 6.5, in terms of number of subjects (Sub), number of different actions (Act), multi-modality (MM), number of RGB (RGB) and depth (Depth) frames, availability

Dataset	Sub	Act	MM	RGB	Depth	P	S
MS COCO [112]	-	-	✗	330K	✗	2D	✗
MPII [2]	-	-	✗	40K	✗	2D	✗
PoseTrack [5]	-	-	✗	22K	✗	2D	✗
MPI-3DHP [126]	8	-	✗	1.3M	✗	3D	✗
3DPW [123]	7	8	✗	51K	✗	3D	✓
Human3.6M [80]	11	15	✗	3.6M	0.45M	3D	✗
PHSPD (ours)	21	31	✓	2.1M	2.1M	3D	✓

Table 6.5: A tally of widely-used human pose and shape datasets.

of annotated poses (P) and shapes (S) for each frame. The datasets, MPII [2], MS COCO [112] and PoseTrack [5], provide only RGB images with 2D pose annotations, where depth images and shape annotations are not considered; MPI-INF-3DHP [126] contains 1.3M in-house frames with 3D pose annotations, but without depth images. 3DPW [123] has 51K frames with extreme pose and SMPL shape annotations, meanwhile it does not possess depth images. Compared with Human3.6M [80], our PHSPD dataset comes with 3D shape annotations, RGB images of higher resolution, more depth images with higher resolution, more subjects, and more action types. More importantly, PHSPD is the only dataset that comes with polarization images for human pose estimation and shape modelling.

6.4 Experiments

Our proposed approach is empirically examined in two major aspects, namely surface normal estimation, and pose and shape estimation from a polarization image. Sec. 6.4.1 focuses on the empirical evaluations of surface normal estimation on the widely used SfP benchmark [9] in shape from polarization, as well as our PHSPD dataset. Sec. 6.4.2 and 6.4.3 evaluate 3D pose estimation on the synthetic SURREAL dataset [199] and the PHSPD dataset, and shape estimation on our PHSPD dataset. Ablation study is presented in Sec. 6.4.4 to analyze the effect of individual components in our approach.

Evaluation Metrics. For surface normal estimation, we report mean angle error (MAE), which measures the angle error between the target and

estimated normal map, formally defined as

$$\text{MAE} = \sum_{x,y} |1 - \langle \hat{\mathbf{m}}_{[x,y]}, \mathbf{m}_{[x,y]} \rangle|, \tag{6.26}$$

where $\hat{\mathbf{m}}$ and $\mathbf{m}$ are predicted and target normal maps and x, y are the pixel indices of the normal map. For human pose and shape estimation, we report the Mean Per Joint Position Error (MPJPE), defined as

$$\text{MPJPE} = \frac{1}{24} \sum_{j=1}^{24} \|\mathbf{J}_{3\text{D}}^{j} - \hat{\mathbf{J}}_{3\text{D}}^{j}\|_2, \tag{6.27}$$

where $\hat{\mathbf{J}}_{3\text{D}}, \mathbf{J}_{3\text{D}} \in \mathbb{R}^{24\times 3}$ are the predicted and target 3D joints. Pelvis-aligned MPJPE (PEL-MPJPE) means MPJPE after root joint alignment, which is defined as

$$\text{PEL-MPJPE} = \frac{1}{24} \sum_{j=1}^{24} \|(\mathbf{J}_{3\text{D}}^{j} - \mathbf{J}_{\text{root},3\text{D}}) - (\hat{\mathbf{J}}_{3\text{D}}^{j} - \hat{\mathbf{J}}_{\text{root},3\text{D}})\|_2, \tag{6.28}$$

where $\mathbf{J}_{\text{root},3\text{D}}$ and $\hat{\mathbf{J}}_{\text{root},3\text{D}}$ are the root joint positions. Procrustes-aligned MPJPE (PA-MPJPE) means MPJPE after aligning the predicted pose with the target by a rigid transformation (R, t), which is defined as

$$\text{PA-MPJPE} = \frac{1}{24} \sum_{j=1}^{24} \|\mathbf{J}_{3\text{D}}^{j} - (\hat{\mathbf{J}}_{3\text{D}}^{j} \cdot R + t)\|_2. \tag{6.29}$$

Note that (R, t) are calculated by minimizing the average distance between the target 3D joints $\mathbf{J}_{3\text{D}}$ and the transformed predicted joints $(\hat{\mathbf{J}}_{3\text{D}} \cdot R + t)$. Percentage of correct key-points (PCK) means the percentage of joints whose PEL-MPJPE is less than 100mm, which is defined as

$$\text{PCK} = \frac{1}{24} \sum_{j=1}^{24} \mathbb{1}(\|(\mathbf{J}_{3\text{D}}^{j} - \mathbf{J}_{\text{root},3\text{D}}) - (\hat{\mathbf{J}}_{3\text{D}}^{j} - \hat{\mathbf{J}}_{\text{root},3\text{D}})\|_2 < 100mm), \tag{6.30}$$

where $\mathbb{1}(\cdot)$ is the indicator function. As for the evaluation of human shape, 3D point to surface error (P2S) is employed, where the iterative closest point (ICP) alignment between predicted body mesh and the ground-truth human body point-cloud is applied. For each vertex of the human body mesh, its closest point in the point-clouds is identified to form a pair, and the average distance of all the pairs is computed as P2S.

Evaluation Datasets. The widely-used SfP dataset [9] is employed to evaluate the performance of our proposed Polar2Normal component. We follow the typical training (236 polarization images with 1224×1024 resolution) and testing (27 images) scheme as in [9], where a 256×256 patch on a image is randomly cropped for training. For testing, 20 overlapped patches in one image are first cropped for evaluation and then fused to form the final predicted normal map. We also demonstrate the effectiveness of our approach on SURREAL [199], a synthetic dataset containing color images rendered from motion-captured SMPL human shapes. Polarization images can be synthesized using color and depth images provided by SURREAL dataset, which will be covered later. We choose subset "run1" and down-sample over time to recruit one from every 20 consecutive frames. Finally, the train set comprises 123,860 samples and test set has 26,650 samples. SURREAL dataset is only used for the evaluation of normal and pose estimation due to the lack of ground truth point-clouds of the clothed human bodies.

A polarization image (polarizers of $0°$, $45°$, $90°$ and $135°$) can be synthesized from the rendered depth and color image in SURREAL dataset. In detail, from the depth image, we obtain the normal map and calculate the corresponding zenith and azimuth angles. From the color image, we get the gray-scale image and take it as the polarization image under $0°$ degree polarizer, denoted by $I(0°)$. Assuming diffuse reflection of the human body surface, the degree of polarization ρ can be calculated according to the equation,

$$\rho = \frac{(n - \frac{1}{n})^2 \sin^2\theta}{2 + 2n^2 - (n + \frac{1}{n})^2 \sin^2\theta + 4\cos\theta\sqrt{n^2 - \sin^2\theta}}, \tag{6.31}$$

with the calculated zenith angle θ for each pixel and refractive index n known as 1.5. Then the upper and lower bound of the illumination intensity I_{max} and I_{min} can be solved in closed form with the constraints as follows,

$$\rho = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}, \tag{6.32}$$

and

$$I(0) = \frac{I_{max} + I_{min}}{2} + \frac{I_{max} - I_{min}}{2}\cos(2\varphi). \tag{6.33}$$

Finally, we can use the equation,

$$I(\phi_{pol}) = \frac{I_{max} + I_{min}}{2} + \frac{I_{max} - I_{min}}{2} \cos(2(\phi_{pol} - \varphi)), \tag{6.34}$$

to get the image under polarizer ϕ_{pol} of 45°, 90° and 135°, or any arbitrary degree. To make it close to the real-world applications, we add Gaussian noise with the standard deviation as $\sigma = 1/255$ to each pixel of the synthetic polarization image and then quantize the intensity value to 8 bits. Due to the fact that we only have geometric information for the human bodies, the synthetic polarization images only have valid values in the human body areas.

Implementation Details. Our real-world PHSPD dataset is involved in the evaluations of all three tasks of normal estimation, pose estimation and clothed shape estimation. The target surface normal is obtained by fusing multi-view normal maps calculated from multiple depth images. For a specific pixel on the polarization image, we find its corresponding position on the multi-view depth images via the calibrated extrinsic parameters between the polarization camera and each depth camera. Then the multiple normal directions are obtained for this pixel and they are averaged together to obtain the final target surface normal map. Subject 4, 7, 11 in PHSPDv1 and subject 1, 2, 7 in PHSPDv2 are chosen to form the test set, resulting in 117,860 samples. The training set consists of the rest 400,785 samples.

For Polar2Normal, each of the encoders and decoders contains 6 sequential blocks, with each block having one up- or down-sampling layer and two convolutional layers. The Polar2Normal model is trained for 600 epochs on SfP dataset, 30 epochs on SURREAL dataset and 8 epochs on PHSPD dataset with Adam optimizer [96]. The learning rate is 0.001, which decays to 0.0001 after training 200 epochs, 10 epochs, and 6 epochs, respectively. The batch size is 16 for training on three datasets. For Polar2Shape model, ResNet50 [73] is used as the backbone CNN model. The extract 1024-dimensional feature is directly regressed to the final outputs: $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, and $\mathbf{t}$. The pose estimation model is trained for 30 epochs on SURREAL dataset and 8 epochs on PHSPD dataset with Adam optimizer [96]. The learning rate is 0.001 and decays to 0.0001 after training 10 epochs and 4 epochs respectively. The batch size for

training is 32. The trade-off parameter α is set to 0.5. The tuning parameters of the loss terms are set to $\lambda_{\text{shape}} = 0.1$, $\lambda_{\text{trans}} = 0.1$ and $\lambda_{\text{joints}} = 10$, respectively. The three weights used in the Polar2Shape stage, namely the normal term λ_n, the depth data term λ_d, and the smoothness term λ_s are empirically set to 1.0, 0.06, and 0.55, respectively.

6.4.1 Evaluation of Surface Normal Estimation

In this task, our approach is compared with three baselines: a conventional method HeightfP [183], a most recent work DeepSfP [9] and our preceding work HumanSfP1 [269]. Compared with HumanSfP2 in Fig. 6.2, HumanSfP1 does not include the computation of coarse normal maps and a further step of normal residual refinement.

From the quantitative results in Tab. 6.6 and the qualitative results in Fig. 6.8, our method consistently outperforms the state-of-the-art SfP methods, HeightfP [183] and DeepSfP [9], for the task of shape from polarization. The poor performance of HeightfP [183] could be attributed to its noise-free assumption that may not hold in the captured images. Though DeepSfP [9] incorporates the ambiguous normal maps as physical priors, the ambiguous normal maps are directly concatenated with the polarization image to form its input, which may overlook the implicit geometric clues. As a result, it performs less well when comparing to our method, especially in complex scenes such as Christmas, Dragon and Horse, where the results of our method achieve $\sim 3^\circ - 5^\circ$ improvement. It is also demonstrated in the visual results of Fig. 6.8. Comparing to our previous work HumanSfP1 [269], as illustrated in Tab. 6.6, our new method, HumanSfP2, exceeds HumanSfP1 in most scenarios, especially Horse and Dragon, where we have $\sim 2^\circ$ improvement. Only for the two scenes of Box and Christmas, the results of both HumanSfP1 and HumanSfP2 are almost identical. The observation demonstrates the advantages of our newly proposed two-step strategy for surface normal estimation.

We also evaluate the normal estimation of human body surface on both SURREAL dataset and our PHSPD dataset. Through both the quantitative results of Tab. 6.7 and the visual results of Fig. 6.9, it is observed that our

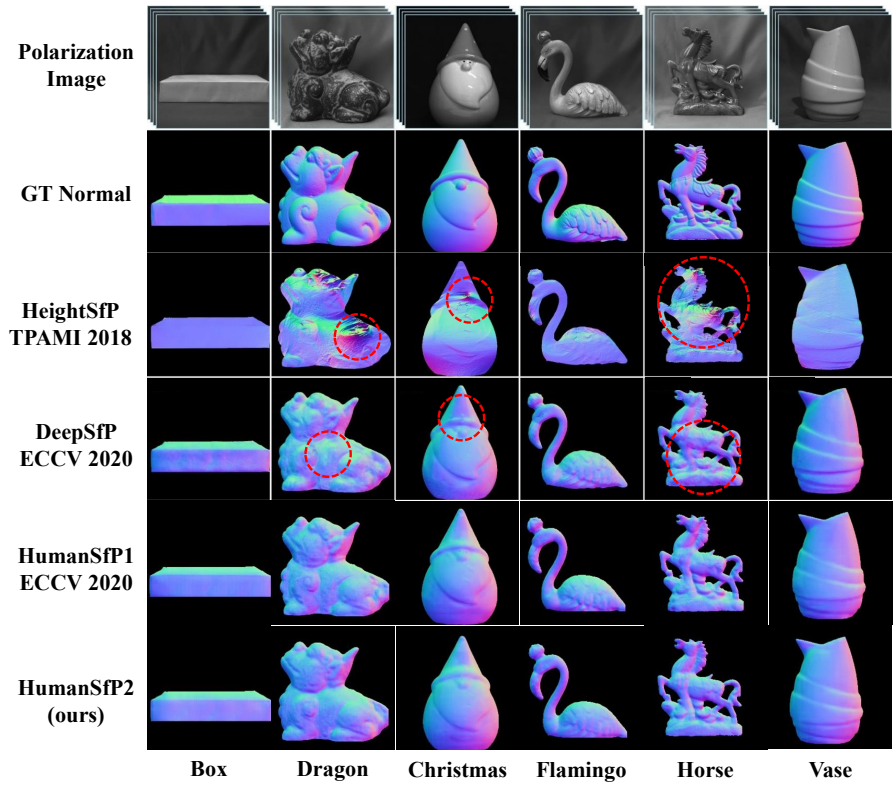


Figure 6.8: Exemplar results of normal map prediction on SfP dataset in terms of MAE.

method consistently outperforms the state-of-the-art surface normal prediction methods, namely HeightfP [183], DeepSfP [9] and HumanSfP1 [269]. Similar to the results presented in Tab. 6.6, the conventional method HeightfP shows least appealing performance, which possibly results from its assumptions of polarization images with high-precision pixel intensity and ideal noise-free environment. This is in contrast to the deep neural network based approaches that are often more robust to environmental noise and polarization images with standard pixel representation. Rather than directly concatenating and feeding the input polarization image and the corresponding ambiguous normal maps into a deep model, HumanSfP1 and HumanSfP2 categorize each pixel into one of the ambiguous normal maps, and obtain a fused normal that incorporates explicit geometric information for normal estimation. This could be the main factor that our methods exceed the state-of-the-art DeepSfP on both SURREAL and PHSPD datasets. Moreover, HumanSfP2 out-performs Hu-

Scene	HeightfP [183]	DeepSfP [9]	HumanSfP1 [269]	HumanSfP2 (ours)
Dragon	49.16	21.55	18.71	**16.88**
Horse	55.87	22.27	21.27	**19.64**
Christmas	39.68	13.50	**8.56**	8.57
Box	31.00	23.31	**21.94**	22.04
Flamingo	36.05	**20.19**	20.51	20.44
Vase	36.88	10.32	9.20	**9.18**
Whole Set	41.44	18.52	17.22	**16.73**

Table 6.6: Quantitative results of surface normal estimation on SfP dataset in terms of MAE.

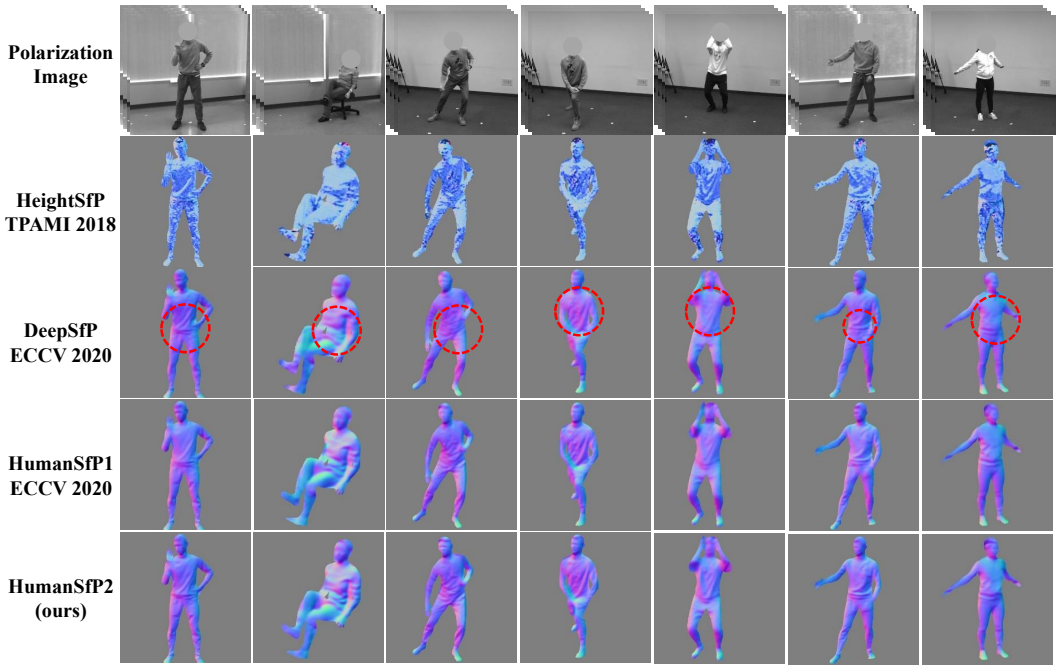


Figure 6.9: Exemplar results of normal map prediction on PHSPD dataset.

manSfP1 over both datasets, which may be attributed to the two-step model proposed in HumanSfP2, where the first step focuses more on the coarse normal map, and the second step pays more attention to the fine normal details. Qualitative results in Fig. 6.9 also demonstrate the superior results of the proposed HumanSfP2.

6.4.2 Evaluation of Pose Estimation

This section concerns the qualitative and quantitative evaluation of the estimated SMPL poses. Our comparison methods consist of HMR [91] and SPIN [99]. Since both HMR and SPIN are trained on single RGB images (C),

Method	SURREAL	PHSPD
HeightfP [183]	20.03	39.95
DeepSfP [9]	7.59	23.08
HumanSfP1 [269]	7.08	22.06
HumanSfP2 (ours)	**6.79**	**21.36**

Table 6.7: Quantitative results of surface normal estimation on SURREAL and PHSPD datasets in terms of MAE.

Method	Input	SURREAL		
		PEL-MPJPE↓	PA-MPJPE↓	PCK↑
HMR [91]	C	135.95	100.66	0.53
SPIN [99]	C	95.08	80.67	0.58
HMR (polar)	P	113.82	86.85	0.55
HumanSfP1 [269]	P	84.78	60.82	0.72
HumanSfP2 (ours)	P	**59.17**	**46.58**	**0.85**

Table 6.8: Quantitative results of human pose estimation on SURREAL dataset.

for PHSPD dataset, images from an RGB camera having similar angle of view with the polarization camera are used as the input for evaluation. In addition to HMR and SPIN, for fair comparison, HMR (polar) is included as another baseline, where HMR model is trained from scratch on the polarization images (P) from either SURREAL dataset or our PHSPD dataset. HumanSfP1 [269] is also employed as another baseline that uses Euclidean distance to measure the distance between the predicted and target poses. We also provide supplementary video[2] for better visualization of our results.

From Tab. 6.8 and 6.9, it is observed that our HumanSfP2 method produces the lowest errors in MPJPE, PEL-MPJPE and PA-MPJPE, and the highest PCK score among all competing methods. Comparing to HMR and SPIN that take RGB images as the input source of data, and HMR (polar) that takes polarization images as the input, our HumanSfP2 out-performs them consistently by a large margin on both SURREAL and PHSPD datasets. Such superior performance lies in two important factors.

The first is the engagement of normal map as part of the input, which will be explained in Sec. 6.4.4 in detail. It is of interest to point out that normal map as part of the input data source is capable of reducing the average joint error by about $10-20$ mm in MPJPE, PEL-MPJPE and PA-MPJPE, while the

[2]Link of supplementary video.

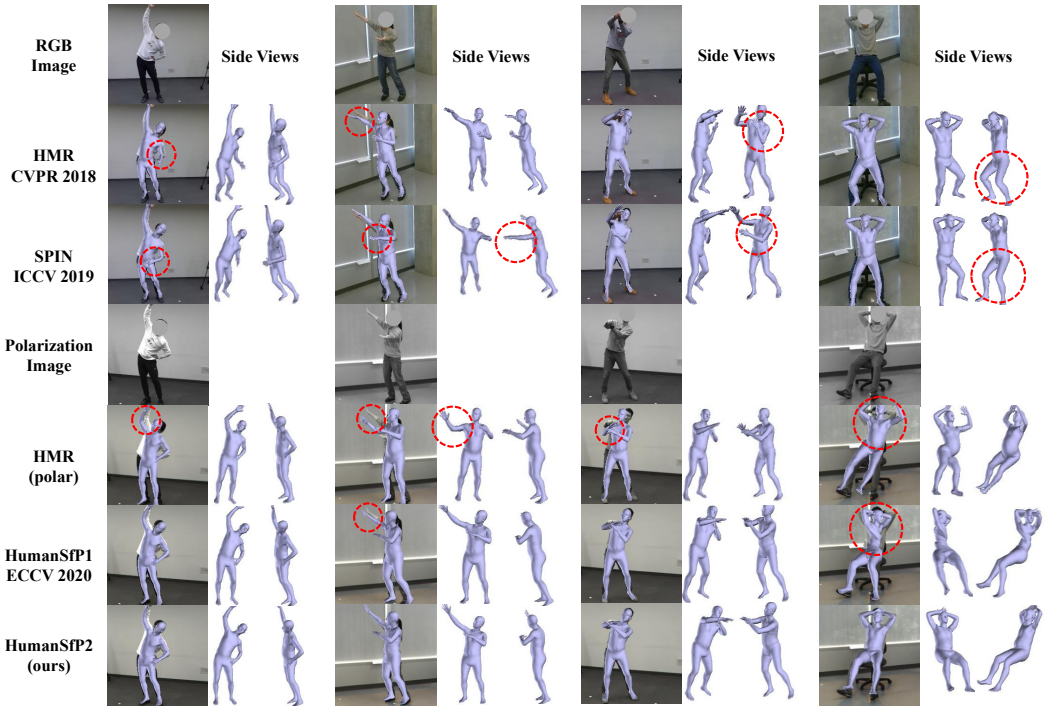


Figure 6.10: Exemplar results of human pose estimation on PHSPD dataset.

Method	Input	PHSPD			
		MPJPE↓	PEL-MPJPE↓	PA-MPJPE↓	PCK↑
HMR [91]	C	-	106.27	68.58	0.57
SPIN [99]	C	-	91.92	49.79	0.64
HMR (polar)	P	113.74	88.24	59.05	0.68
HumanSfP1 [269]	P	66.97	63.12	42.08	0.83
HumanSfP2 (ours)	P	**62.04**	**54.98**	**36.88**	**0.88**

Table 6.9: Quantitative results of human pose estimation on PHSPD dataset.

three competing methods do not include the normal clues in pose estimation.

The second factor is the introduction of geodesic loss to measure the predicted and target SMPL poses. Comparing to HumanSfP1, our approach improves the joint error by about 5mm and PCK by 0.05 in PHSPD dataset, and 15mm and 0.13 in SURREAL dataset. The quantitative results illustrate the effectiveness of leveraging geodesic distance in pose estimation. Visual results in Fig. 6.10 also demonstrate the effectiveness of our HumanSfP2 approach, where the input RGB and polarization images are shown in the 1st and 4th row and poses from two side views are also presented to better evaluate the predicted results. HumanSfP1 may predict unnatural poses as the Euclidean distance is not suitable in measuring joint displacements due to relative joint

Method	Input	P2S
DepthHuman [192]	C	83.86
PIFuHD [173]	C	67.05
PIFu [172]	C	62.13
HMD [264]	C	43.72
PaMIR [257]	C	47.24
HumanSfP2 (initial)	P	39.79
HumanSfP2 (ours)	P	**38.24**

Table 6.10: Quantitative results of clothed human shape estimation.

rotations.

6.4.3 Evaluation of Shape Estimation

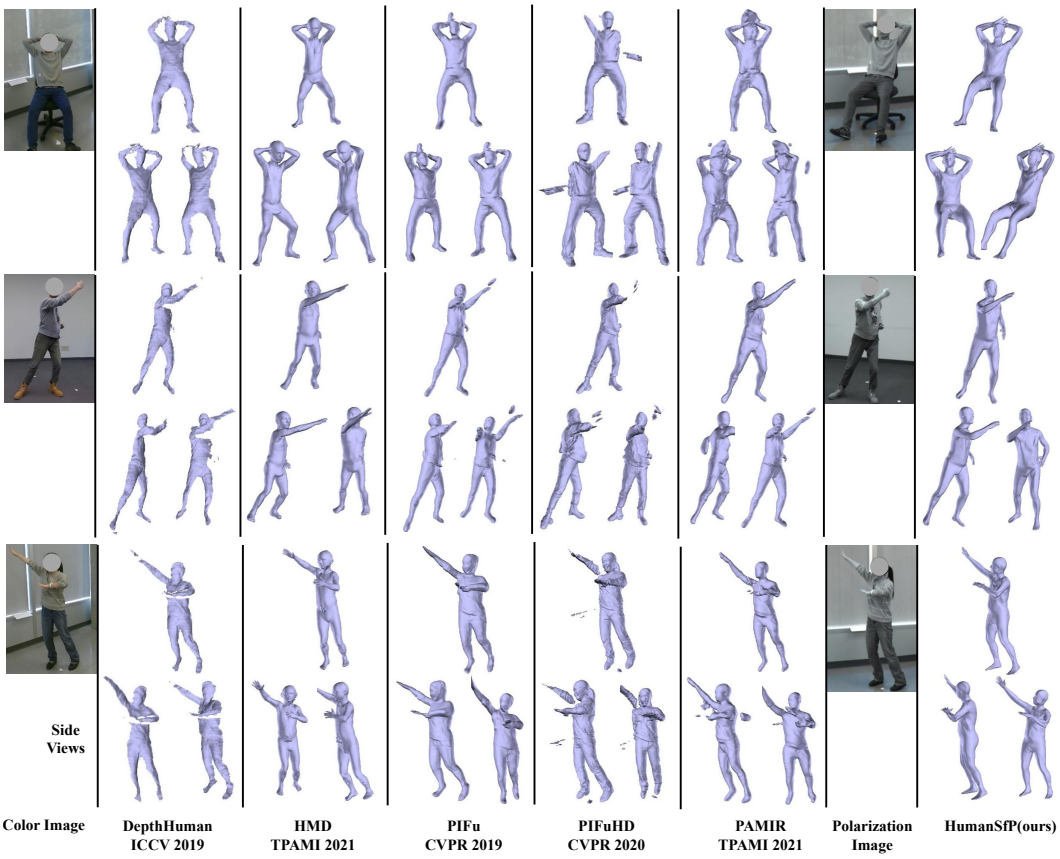


Figure 6.11: Exemplar estimation results of clothed body shapes.

Considering there is no previous work in this new task of single polarization image based clothed human shape reconstruction, four RGB-based methods are recruited for comparison. They are DepthHuman [192], HMD [264], PIFu [172], PIFuHD [173] and PaMIR [257], where RGB image having closest

Figure 6.12: Exemplar estimation results of clothed body shapes, obtained on polarization images from new scene context.

view with polarization image is used as the input for evaluation. The estimated clothed shape is then compared with human point-cloud to calculate 3D point to surface error (P2S).

Quantitative results are displayed in Tab. 6.10. Here, DepthHuman performs the worst, which may be partly attribute to its consideration of the surface depth instead of the entire body shape. PIFu [172] and PIFuHD [173] show similar results, with PIFuHD having slightly larger error. The explanation is PIFu requires human mask as a prior, while PIFuHD does not have such assumption. However, both methods do not take human pose into consideration when predicting their implicit surfaces. The 3D error from HMD [264] is relatively small, possibly due to the accurate initial shape estimation. Our SfP approach achieves the best performance, which should be credited to its exploitation of the estimated normal maps.

Exemplar visual results are presented in Fig. 6.11. It is observed that for DepthHuman, only a partial mesh with respect to the view in the input image is produced, as is also evidenced in Tab. 6.10. HMD, on the other hand, does not work well, as evidenced by the often error-prone surface details. This may be attributed to the less reliable shading representation, given the new environmental lighting and texture ambiguities existed in these RGB images. PIFu [172] and PIFuHD [173] are able to predict human with clothing details, but both suffer from the relative inaccurate pose inference, making their results ill-aligned with the subject in the image space. PaMIR [257] estimates reasonable poses but reconstructs in-accurate human shapes, which could attribute to the in-correct information provided by its deep implicit function. Our HumanSfP approach is shown capable of producing reliable prediction of

Method	SURREAL	PHSPD
HumanSfP2 (color)	13.44	24.79
HumanSfP2 (no-prior)	12.16	24.75
HumanSfP2 (ours)	**6.79**	**21.36**

Table 6.11: Ablation study of our normal estimation component on SURREAL and PHSPD datasets.

Method	Input	SURREAL		
		PEL-MPJPE ↓	PA-MPJPE ↓	PCK ↑
HumanSfP1 [269]	Polar	99.33	70.19	0.65
	Polar+Mask	95.28 (4.05)	67.84 (2.35)	0.67 (0.02)
	Polar+Normal	84.78 (**14.55**)	60.82 (**9.37**)	0.72 (**0.07**)
HumanSfP2 (color)	Color	96.27	74.88	0.69
	Color+Mask	84.84 (11.43)	62.74 (12.14)	0.73 (0.04)
	Color+Normal	80.28 (**15.99**)	59.39 (**15.49**)	0.76 (**0.07**)
HumanSfP2 (ours)	Polar	74.12	56.39	0.78
	Polar+Mask	72.20 (1.92)	55.48 (0.91)	0.78 (0)
	Polar+Normal	59.17 (**14.95**)	46.58 (**9.81**)	0.85 (**0.07**)

Table 6.12: Ablation study of hybrid input in human pose estimation on SURREAL dataset.

clothed body shapes, which again demonstrates the applicability of polarization imaging in shape estimation, as well as the benefit of engaging the surface normal maps in our approach.

Qualitative results presented in Fig. 6.12 showcase the generalization ability of our approach. Note the polarization images are acquired at different physical locations with distinct background scenes that are very dissimilar to those in the training images.

6.4.4 Ablation Study

Polarization and RGB modalities for normal estimation. Here we want to compare the performance of normal map reconstruction from RGB images vs. polarization images. Let HumanSfP2 (color) denote the model that uses only RGB image, HumanSfP2 (no-prior) be the model without incorporating the ambiguous normal maps as the physical priors and with only the polarization image as input. The quantitative results are summarized in Tab. 6.11. We observe that HumanSfP2 (ours) exceeds HumanSfP2 (color) by $\sim 7^\circ$ in SURREAL dataset and $\sim 3.5^\circ$ in PHSPD dataset. The larger improvement of MAE in SURREAL dataset may be the fact that SURREAL

Method	Input	PHSPD			
		MPJPE ↓	PEL-MPJPE ↓	PA-MPJPE ↓	PCK ↑
HumanSfP1 [269]	Polar	85.74	83.70	54.90	0.72
	Polar+Mask	75.76 (9.98)	73.04 (10.66)	50.59 (4.31)	0.77 (0.05)
	Polar+Normal	66.97 (**18.77**)	63.12 (**20.58**)	42.08 (**12.82**)	0.83 (**0.11**)
HumanSfP2 (color)	Color	88.55	77.97	54.32	0.74
	Color+Mask	80.79 (7.76)	71.58 (6.39)	44.55 (9.77)	0.80 (0.06)
	Color+Normal	76.34 (**12.21**)	68.23 (**9.74**)	41.95 (**12.37**)	0.84 (**0.10**)
HumanSfP2 (ours)	0.78	70.73	63.82	44.05	0.83
	Polar+Mask	68.95 (1.78)	60.28 (3.54)	41.58 (2.47)	0.85 (0.02)
	Polar+Normal	62.04 (**8.69**)	54.98 (**8.84**)	36.88 (**7.17**)	0.88 (**0.05**)

Table 6.13: Ablation study of hybrid input in human pose estimation on PHSPD dataset.

dataset is synthesized from naked and minimal dressed body shapes, such that the normal maps of human body are smooth and are relatively easier to predict than the real-world PHSPD dataset. Moreover, similar MAE results are presented by HumanSfP2 (no-prior) and HumanSfP2 (color), which showcases the importance of ambiguous normal maps in our approach that carries the critical geometric clues for high performance in surface normal estimation from polarization images.

Normal maps for pose estimation. This section demonstrates the significant performance gain that normal maps provides in pose estimation. As in Tab. 6.12 and 6.13 with round bracket showing the improvement over the case of polar/RGB as the sole input, the three methods of HumanSfP1[269], HumanSfP2 (color) and HumanSfP2 (ours) are equipped with different input combinations: polar/RGB image, polar/RGB image with foreground mask, and polar/RGB image with predicted normal map. Within each method, the performance gain is particularly significant when normal map is incorporated as input. This may be attributed to the rich geometric information encoded in the normal map representation. Less significant gain is obtained when only mask is incorporated as input, which further demonstrates the effectiveness of geometric information in pose estimation. When comparing across polarization and RGB modalities in HumanSfP2 (ours) and HumanSfP2 (color), there is still noticeable improvement in HumanSfP2 (color), which combines RGB image with normal map as the input. However, the overall performance of

HumanSfP2 (color) is worse than that of HumanSfP2 (ours). The explanation is that the normal maps estimated from RGB images are not as reliable as those obtained from the polarization image counterparts.

6.5 Conclusion

We tackle in this chapter a new problem of estimating human shapes from single 2D polarization images. Our work exemplifies the applicability of engaging polarization cameras as a promising alternative to the existing imaging modalities for human pose and shape estimation. Moreover, by exploiting the rich geometric details in the surface normal of the input polarization images, our SfP approach is capable of reconstructing human body shapes of surface details. We expect this could be a useful tool in many downstream applications.

Chapter 7

Point-based Clothed Human Modeling via Diffusion Models

7.1 Introduction

Clothed human modeling aims to learn clothing deformation dynamics from a set of 3D point-clouds or meshes of clothed human body, facilitating the generation of naturalistic clothing details in target motion animations. This task is inherently challenging owing to the the variety of clothes and human motions. Traditional methods typically employ either basic rigging-and-skinning techniques [11], [114] or rely on physics-based simulations [152], [195], which requires intensive computations and specialized expertise to create a simulation-ready clothing mesh. In contrast, recent data-driven approaches [32], [40], [68], [83], [108], [111], [119], [121], [174], [224], [244], [256], either using implicit or explicit representations, have yielded promising results in this field of research.

Notably, multiple studies [111], [118], [119], [121], [244] have demonstrated the efficacy of point-based representation of clothing shapes, attributed to the compactness and topological flexibility of point-clouds. POP [121] stands out as a pioneer in modeling pose-dependent clothed humans using point-clouds, showcasing the capability of a singular model to manage arbitrary clothes. Subsequent efforts, like FITE [111] and SkiRT [119], adopt a coarse-to-fine strategy, leveraging implicit or explicit techniques to reconstruct a coarse template first. The latest work, CloSET [244], learns pose features directly from the continuous body surface, aiming to address the discontinuity presented by

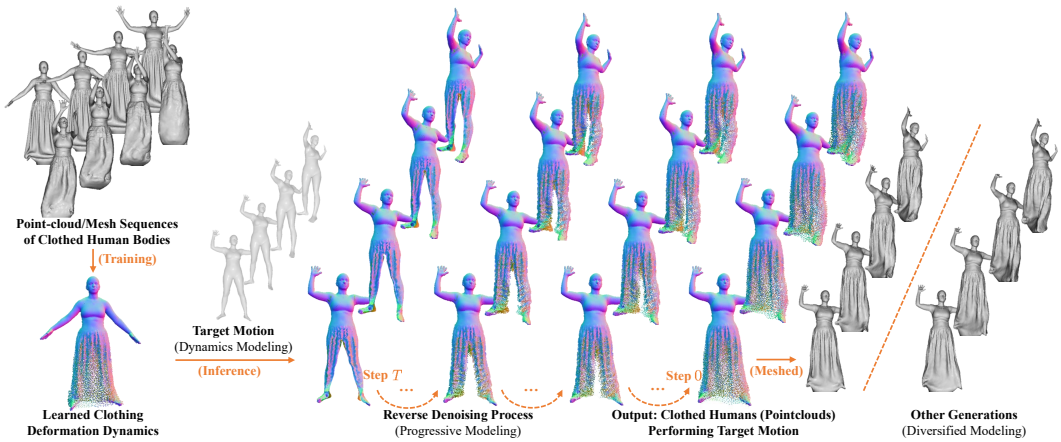


Figure 7.1: Progressive modeling of clothed humans performing a target motion via diffusion models.

the UV map used in earlier research [118], [121].

Despite the encouraging achievements, there are still unresolved challenges in this field of study. The first challenge resides in the *dynamics modeling* of clothed humans, where the clothing deformations are supposed to be nature and smooth, both spatially and temporally, as a person performs various motions. However, existing learning-based approaches [32], [111], [118], [119], [121], [174], [244] focus on the clothing deformations associated with a single pose only, overlooking the underlying correlation and continuity of clothing deformations in a motion sequence. The second challenge relates to the *progressive modeling* of clothed humans, a process that mirrors the iterative refinement typically seen in artifact creation. Earlier works either model clothed humans in a single step [121], [174], [244], or employ a two-step coarse-to-fine strategy [111], [119], thereby missing the opportunity to fully exploiting the benefits of progressive refinement in modeling clothes. The third challenge lies in the *diversified modeling* of clothed humans, which is in accord with the real-world observation that identical outfits and motions can yield varying patterns of cloth wrinkles. Existing methods [111], [119], [121], [174], [244] are mostly deterministic, thereby limiting the range of variations in response to specific outfits and motions.

To address these challenges, we propose ClothDiffuse, a diffusion-based method that learns the dynamics of clothing deformations for the realistic gen-

eration of clothing details in target motion animations. *Our key insight is to involve all the three significant aspects in our framework: dynamics modeling, progressive modeling, and diversified modeling of clothed humans.* Specifically, we take sequential frames of unclothed posed bodies, such as SMPL body shapes, as the input. The dynamic features of vertices on the input bodies, including 3D positions, velocities and accelerations, are mapped to UV positional maps and then processed through a 3D CNN to encode the dynamic features of the motion sequence. Simultaneously, we utilize a learnable tensor to represent pixel-aligned garment features. For a specific query point on the unclothed bodies, we sample the respective dynamic motion feature and garment feature. The shape decoder, conditioned on these features, progressively denoises a Gaussian noise for multiple steps to yield the final prediction that includes motion-dependent clothing wrinkle displacements and the normal direction of the query point, referred to as *dynamic and progressive modeling*. The stochastic nature of sampling process results in a diverse but genuine outcomes for each inference, termed as *diversified modeling*. After applying local transformation and dense querying on the input unclothed bodies, we achieve the generation of point-based clothed humans in accord with the target motion.

Our contributions are summarized as follows:

- We propose ClothDiffuse, a diffusion-based approach designed to learn the dynamics of clothing deformations. This enables the realistic generation of clothing details in animations with target motions, focusing on motion-dependent dynamics rather than pose-dependent static features.

- To our knowledge, we are the first learning-based approach that involves all the three critical aspects in clothed human modeling: dynamics modeling, progressive modeling, and diversified modeling. This unique combination allows for a more nuanced and effective representation of clothing deformation dynamics. Empirical experiments demonstrate that our approach outperforms multiple baselines on two challenging benchmarks.

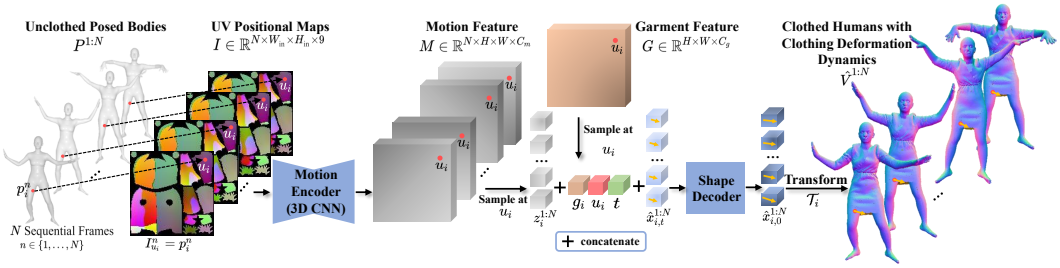


Figure 7.2: Overview of our approach for clothed human modeling.

7.2 Method

Our objective is to model clothed humans given N consecutive frames of posed but unclothed human bodies as input. This model is trained with a set of 3D point-clouds or mesh sequences of clothed human bodies. We assume the corresponding fitted or registered unclothed bodies, such as SMPL [116] or SMPL-X [153], are known, in accordance with prior works [111], [119], [121], [244]. The overview of our approach is illustrated in Fig. 7.2. We begin by introducing the motion-dependent feature encoding in Sec. 7.2.1, then proceed to detail the progressive modeling of clothing using diffusion models in Sec. 7.2.2, and finally outline the procedure of training and inference in Sec. 7.2.3.

7.2.1 Motion-Dependent Feature Encoding

Pose Encoder. We use $V^{1:N} = \{v_i^{1:N}\}_{i=1}^{M_s}$ to indicate a sequence of N frames' 3D point-clouds of clothed bodies with the n as the frame index. The corresponding fitted or registered unclothed bodies are represented as $P^{1:N}$. Using the unclothed bodies $P^{1:N}$ as input, we first map the dynamic features of each point $p_i^n \in \mathbb{R}^9$ on the surface to the corresponding 2D position $u_i \in \mathbb{R}^2$ on a UV positional map, denoted as $I^n_{u_i} = p^n_i$. These dynamic features include 3D position, velocity and acceleration. Then we obtain the UV positional maps $I \in \mathbb{R}^{N\times H_{\text{in}}\times W_{\text{in}}\times 9}$ and forward to the a 3D CNN as encoder to extract the dynamic features of the motion sequence, $M \in \mathbb{R}^{N\times H\times W\times C_m}$. Meanwhile, we employ $G \in \mathbb{R}^{H\times W\times C_g}$ as a learnable variable to encode pixel-aligned garment feature.

Existing works [111], [119], [121], [244] mainly focus on the pose-dependent modeling of clothed humans, overlooking the correlation and continuity of pose features for a motion sequence. In contrast, we propose to encode motion-dependent features directly, aiming for enhanced modeling of clothed humans in a motion. Furthermore, as the UV map of template body maintains a fixed topology, the 3D CNN encoder is able to consistently extract vertex-aligned motion features, where the vertex p_i^n in each frame is mapped to an identical position u_i on the UV map. While the PointNet-based encoder in [244] permits continuous surface sampling, its dynamics extension [54] may struggle to robustly extract temporal information in 3D space, particularly for fast-moving limbs. This limitation arises because the point aggregation in PointNet does not take into account topology constraints.

7.2.2 Diffusion-based Cloth Modeling

Diffusion Models. To be general, we represent $x_0^{1:N}$ as a N-frame data sample drawn from the true data distribution. In the forward diffusion process, we add small amount of Gaussian noise to the sample over T steps, generating a sequence of noisy samples denoted as $\{x_t^{1:N}\}_{t=1}^T$. This process is described by

$$q(x_t^{1:N}|x_{t-1}^{1:N}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}^{1:N}, \beta_t I), \tag{7.1}$$

where the step sizes are controlled by a variance schedule $\{\beta_t \in (0,1)\}_{t=0}^T$ and I represents the identity matrix. When the number of steps T is sufficiently large, $x_T^{1:N}$ approaches a normal distribution, $\mathcal{N}(0, I)$. As noted in [75], for a specific diffusion step t, instead of repeatedly adding noises to $x_0^{1:N}$, we can directly derive $x_t^{1:N}$ through

$$x_t^{1:N} = \sqrt{\bar{\alpha}_t}x_0^{1:N} + \sqrt{1-\bar{\alpha}_t}\epsilon,\ \epsilon \sim \mathcal{N}(0, I), \tag{7.2}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{k=0}^t \alpha_k$.

The reverse process aims to reconstruct a true data sample from Gaussian noise $x_T^{1:N} \sim \mathcal{N}(0, I)$, utilizing a neural model f_θ that progressively denoises

$x_T^{1:N}$ over T steps. This allows us to sample from the learned data distribution by gradually denoising a Gaussian noise. Formally, the process is defined as:

$$p_\theta(x_{t-1}^{1:N}|x_t^{1:N}, x_0^{1:N}) = \mathcal{N}(\mu_\theta(x_t^{1:N}, x_0^{1:N}, t, y), \tilde{\beta}_t I), \tag{7.3}$$

where t is the diffusion step, y encompasses prior conditions, such as text, category label and image, and $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. According to previous work [75], [164], $f_\theta(x_t^{1:N}, x_0^{1:N}, t, y)$ can be defined in three different ways to recover μ_θ: (i) the denoised sample at $t-1$ directly, $f_\theta = \hat{x}_{t-1}^{1:N}$, (ii) the Gaussian noise, $f_\theta = \hat{\epsilon}$, or (iii) the unnoised sample, $f_\theta = \hat{x}_0^{1:N}$. Subsequently, the training of f_θ is to minimize corresponding loss function, expressed as:

$$\mathcal{L} = \begin{cases} \mathbb{E}_{t\in[1,T], x_{t-1}^{1:N}\sim q(x_{t-1}^{1:N})}[\|x_{t-1}^{1:N} - \hat{x}_{t-1}^{1:N}\|] & \text{(i)}, \\ \mathbb{E}_{t\in[1,T], \epsilon\sim\mathcal{N}(0,I)}[\|\epsilon - \hat{\epsilon}\|] & \text{(ii)}, \\ \mathbb{E}_{t\in[1,T], x_0^{1:N}\sim q(x_0^{1:N})}[\|x_0^{1:N} - \hat{x}_0^{1:N}\|] & \text{(iii)}. \end{cases} \tag{7.4}$$

During inference, μ_θ can be obtained as follows:

$$\mu_\theta = \begin{cases} \hat{x}_{t-1}^{1:N} & \text{(i)}, \\ \frac{1}{\sqrt{\alpha_t}}(x_t^{1:N} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}) & \text{(ii)}, \\ \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t^{1:N} + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\hat{x}_0^{1:N} & \text{(iii)}. \end{cases} \tag{7.5}$$

In this work, we employ method (iii), which directly predicts the unnoised sample. This approach has been shown in [164] to yield better performance. Then we can obtain μ_θ as follows:

$$\mu_\theta = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t^{1:N} + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\hat{x}_0^{1:N}, \tag{7.6}$$

where $f_\theta = \hat{x}_0^{1:N}$.

Shape Decoder. In our scenario, the term $x_{i,t}^{1:N}$ in Eq. (7.1) consists of motion-dependent wrinkle displacements $r_i^{1:N} \in \mathbb{R}^3$ and the normal direction $n_i^{1:N} \in \mathbb{R}^3$ of $p_i^{1:N}$ on the surface of unclothed posed bodies. To reconstruct $x_{i,0}^{1:N}$, we first extract the corresponding dynamic features $z_i^{1:N} \in \mathbb{R}^{N\times C_m}$ from M and the garment feature $g_i \in \mathbb{R}^{C_m}$ from G at u_i on each temporal frame. Combined with the sampling position u_i, these features form the prior conditions defined in Eq. (7.3), denoted as $y = [z_i^{1:N}, g_i, u_i]$. At diffusion step t, our

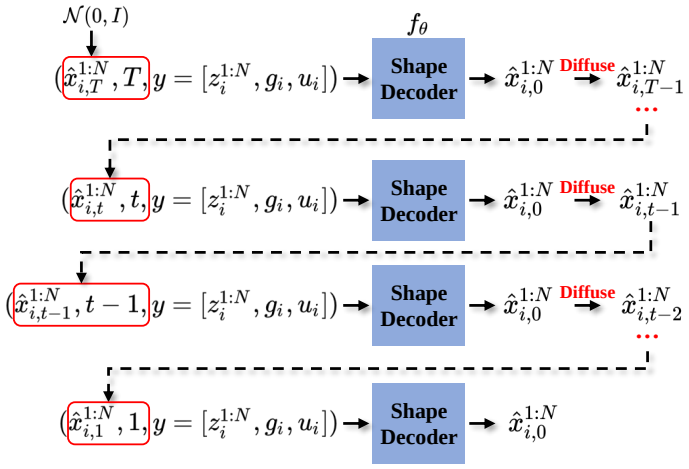


Figure 7.3: Process of inference via our diffusion-based model.

shape decoder is trained to directly predict unnoised sample $x_{i,0}^{1:N}$, expressed by

$$\hat{x}_{i,0}^{1:N} = f_\theta(\hat{x}_{i,t}^{1:N}, t, y). \tag{7.7}$$

During the stage of inference, the shape decoder progressively denoise random Gaussian noise $\hat{x}_{i,T}^{1:N}$ to produce the final prediction, $\hat{x}_{i,0}^{1:N} = [\hat{r}_i^{1:N}, \hat{n}_i^{1:N}]$. Details regarding training and inference are provided in the subsequent Sec. 7.2.3. After getting wrinkle displacements and the normal direction of $p_i^{1:N}$, the local transformation $\mathcal{T}_i$ is applied to derive the clothing points for the query points $p_i^{1:N}$ across the motion sequence, expressed by

$$\hat{v}_i^{1:N} = \mathcal{T}_i \cdot \hat{r}_i^{1:N} + p_i^{1:N}. \tag{7.8}$$

Through the dense querying on the posed but unclothed bodies, we achieve the generation of clothed humans with motion-dependent clothing deformations, represented as $\hat{V}^{1:N} = \{\hat{v}_i^{1:N}\}_{i=1}^{M_p}$.

7.2.3 Training and Inference

Training. To train our diffusion-based model, following [75], [164], we first uniformly sample a diffusion step t and obtain the corresponding noisy sample $x_{i,t}^{1:N}$. Meanwhile, we obtain the conditions $y = [z_i^{1:N}, g_i, u_i]$ for the query points $p_i^{1:N}$. After feeding them to the shape decoder and applying local transformation, we obtain the final outputs, including motion-dependent wrinkle

displacements $\hat{r}_i^{1:N}$, the normal direction $\hat{n}_i^{1:N}$, and clothes points $\hat{v}_i^{1:N}$ over the sequence.

The training loss is defined as

$$\mathcal{L} = \lambda_{\mathrm{c}}\mathcal{L}_{\mathrm{c}} + \lambda_{\mathrm{n}}\mathcal{L}_{\mathrm{n}} + \lambda_{\mathrm{d}}\mathcal{L}_{\mathrm{d}} + \lambda_{\mathrm{r}}\mathcal{L}_{\mathrm{r}} + \lambda_{\mathrm{g}}\mathcal{L}_{\mathrm{g}}, \tag{7.9}$$

where the $\lambda_\cdot$ are the weights to balance each loss. Specifically, $\mathcal{L}_{\mathrm{c}}$ is the normalized Chamfer Distance that measures the average bi-directional squared distances between the predicted pointclouds $\hat{V}^{1:N}$ and ground-truth $V^{1:N}$, which is defined as:

$$\mathcal{L}_{\mathrm{c}} = \frac{1}{N}\sum_{k=1}^{N}\mathcal{L}_{\mathrm{c}}^k, \tag{7.10}$$

$$\mathcal{L}_{\mathrm{c}}^k = \frac{1}{M_s}\sum_{i=1}^{M_s}\min_j \|v_i^k - \hat{v}_j^k\|_2^2 + \frac{1}{M_p}\sum_{j=1}^{M_p}\min_i \|v_i^k - \hat{v}_j^k\|_2^2.$$

$\mathcal{L}_{\mathrm{n}}$ is the average L_1 distance of normal between each predicted point and its nearest neighbor in the ground-truth point-clouds:

$$\mathcal{L}_{\mathrm{n}} = \frac{1}{NM_p}\sum_{k=1}^{N}\sum_{i=1}^{M_p}\|\hat{n}_i^k - n_j^k\|_1, \tag{7.11}$$

where $j = \arg\min_{v_j^k \in V^k}\|\hat{v}_i^k - v_j^k\|_2$. $\mathcal{L}_{\mathrm{d}}$ is the dynamic loss to regularize the acceleration of predicted point-clouds in a sequence:

$$\mathcal{L}_{\mathrm{d}} = \frac{1}{(N-2)M_p}\sum_{i=1}^{M_p}\sum_{k=2}^{N-1}\|(\hat{r}_i^k - \hat{r}_i^{k-1}) - (\hat{r}_i^{k+1} - \hat{r}_i^k)\|_2^2. \tag{7.12}$$

The last two losses, $\mathcal{L}_{\mathrm{d}}$ and $\mathcal{L}_{\mathrm{g}}$, are the regularization of the norm for the predicted displacements and garment feature respectively, which is described by

$$\mathcal{L}_{\mathrm{r}} = \frac{1}{NM_p}\sum_{k=1}^{N}\sum_{i=1}^{M_p}\|\hat{r}_i^k\|_2^2, \quad \mathcal{L}_{\mathrm{g}} = \frac{1}{HWC_g}\|G\|_2^2. \tag{7.13}$$

Inference. The process of inference is to sample from the learned distribution f_θ by iteratively denoising a Gaussian noise $\mathcal{N}(0, I)$. As is illustrated in Fig. 7.3, at the denoising step t, we first predict the unnoised sample $\hat{x}_{i,0}^{1:N} = f_\theta(x_{i,t}^{1:N}, t, y)$, and then diffuse it to $\hat{x}_{i,t-1}^{1:N}$, which will be the input of f_θ at next step. After T steps of denoising, we get the final output $\hat{x}_{i,0}^{1:N}$.

Benefits. Our method enjoys two significant benefits compared with previous works [111], [119], [121], [244]: *1) Progressive refinement*: The sampling process takes T steps to yield the final prediction, imitating the process of iterative refinement typically seen in artifact creation. This process prioritizes the broader outline initially and refines intricate details subsequently, which is known to produce the final results with improved details than prior studies that aim at modeling clothed human in a single step. *2) Diversity in cloth modeling*: Our approach not only surpasses prior techniques that solely offered deterministic clothed human inferences but also captures a realistic diversity in clothes modeling. The inherent stochastic nature of our sampling process means that each inference produces a diverse and also genuine outcome. This is particularly in accord with the real-world observation that the same outfit and motion can present varied cloth wrinkle patterns. In addition, our approach also allows deterministic modeling of motion-dependent clothing by directly utilizing the Gaussian mean values at every diffusion step in Eqs. (7.2) and (7.3) to bypass random sampling from the Gaussian distributions.

Unseen Clothes Modeling. Our approach is also able to tackle unseen clothes, following the similar scheme described in [121]. Given a sequence of raw scans, we fixed the parameters of pose encoder and shape decoder, and optimize an initialized garment feature tensor to minimize the loss defined in Eq. (7.9). Loss masks are applied when the sequence length is less than N.

7.3 Experiments

Datasets. We evaluate our method and compare with baselines on two commonly used datasets. CAPE [120] is a captured real dataset containing clothed human scans under a variety of motions. We follow [120] to split training and test sets from 3 subjects (00096, 00215, 03375) in 14 different outfits. ReSynth [121] is a synthetic dataset with a larger variation in outfit shapes and motions. We follow [121] for training and test sets split.

Implementation Details. The length of input sequence N is 8 and the input positional maps are of size 128×128. The pose encoder in our model is

Methods	Outfits																	
	anna-001		beatrice-025		christine-027		janett-025		felice-004		carla-004		alexandra-006		eric-035		all	
	knee dress short sleeve		knee dress long sleeve		knee dress short sleeve		short skirt long sleeve		long dress tank top		puffy jacket long pants		loose blouse long pants		blazer jacket long pants		-	
	CD	NML	CD	NML	CD	NML	CD	NML	CD	NML	CD	NML	CD	NML	CD	NML	CD	NML
SCALE [120]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1.49	1.04
SCANimate [174]	1.34	1.35	0.74	1.33	3.21	1.66	2.81	1.59	20.79	2.94	0.90	1.52	2.28	1.84	2.54	1.94	4.30	1.77
POP [121]	0.62	0.82	0.34	**0.75**	1.72	**0.97**	1.24	0.89	7.34	1.24	0.51	**1.02**	1.71	**1.29**	1.34	1.16	1.36	1.02
SkiRT [119]	0.58	**0.81**	**0.31**	0.77	1.54	0.99	1.10	0.82	6.45	1.25	0.48	1.06	1.51	**1.29**	1.30	1.17	-	-
CloSET [244]	-	-	-	-	1.49	**0.97**	-	-	6.01	1.16	0.49	1.04	-	-	-	-	-	-
Ours	**0.54**	**0.81**	0.33	0.76	**1.30**	**0.97**	**1.03**	**0.81**	**5.58**	**1.14**	**0.47**	1.05	**1.50**	**1.29**	**1.29**	**1.15**	**1.13**	**1.01**

Table 7.1: Quantitative results on ReSynth dataset across diverse clothes.

Methods	Outfits					
	blazerlong		shortlong		all	
	CD	NML	CD	NML	CD	NML
SCALE [120]	1.07	1.22	0.89	1.12	-	-
POP [121]	0.78	1.29	0.57	1.24	0.59	1.11
CloSET [244]	0.71	1.15	0.54	1.09	-	-
Ours	**0.68**	**1.12**	**0.49**	**1.08**	**0.54**	**1.09**

Table 7.2: Quantitative results on CAPE dataset.

a 7-layer 3D UNet while shape decoder consists of 8-layer MLPs. The motion and garment feature sizes, C_m and C_g, are set to be 64. Note that when only pose-dependent features are considered, *i.e.*, $N = 1$, our model has comparable parameters with prior works [121], [244]. The predefined querying positions on the motion and garment features are on the grid of size 256×256, resulting in around 43K points for each temporal frame. We follow [133] to encode displacements $r_i^{1:N}$, normal $n_i^{1:N}$ and sampling position u_i with frequency being 6, as well as diffusion step t with frequency being 16. For both datasets, the number of diffusion steps is 100 and linear variance schedule of β is used. Our model is trained with uniformly sampled diffusion step for 90 epochs for both datasets on a single A100 GPU. The learning rate is 0.0001 and decay by 0.01 after training 70 epochs. The batch size during training is 4 for $N = 8$ and 8 for $N = 1$.

Evaluation Metrics. Following previous works [119], [121], [244], we report the averaged Chamfer Distance (CD) defined in Eq. (7.10) and the averaged L_1 normal distance (NML) defined in Eq. (7.11) across all test samples in the unit of $\times 10^{-4} m^2$ and $\times 10^{-1}$ respectively. To holistically reflect the performance of our model on different outfits, especially on challenging ones,

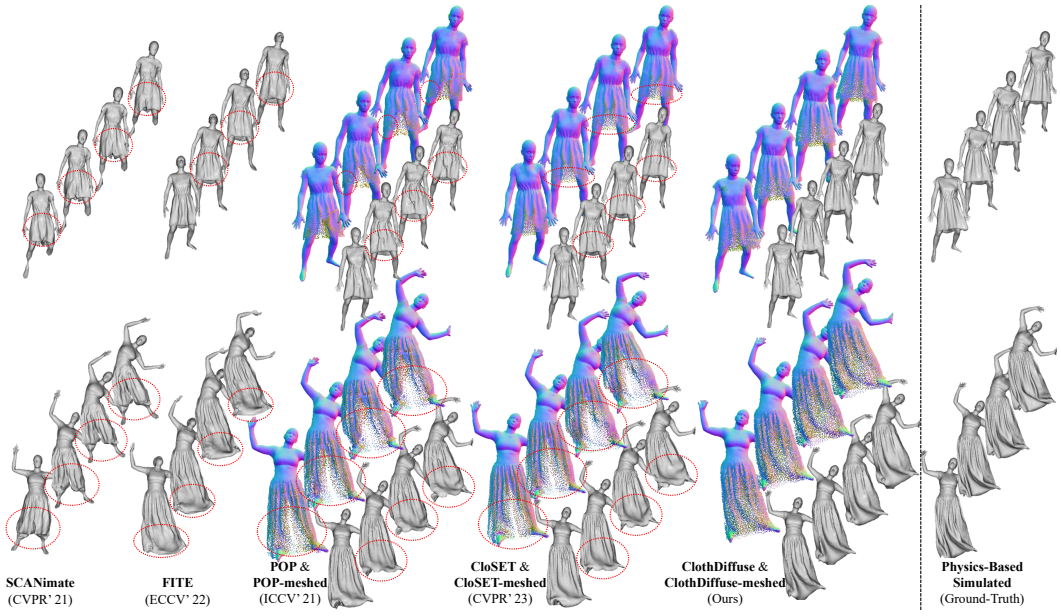


Figure 7.4: Qualitative results on ReSynth dataset.

we list the results for each subject that wears different outfits. We also report Dynamic Errors (DE) as defined in Eq. (7.12) in the unit of $\times 10^{-4} m^2$ to measure the smoothness of predicted point-based clothed human in a motion.

7.3.1 Comparison with State-of-the-Art Methods

We compare our approach with recent state-of-the-art methods: one implicit approach SCANimate [174], and four point-based approaches SCALE [120], POP [121], SkiRT [119] and CloSET [244].

We present the quantitative results on the ReSynth dataset in Tab. 7.1, categorized by different outfits. The results reveal that all four recent point-based methods outperform the implicit method, SCANimate [174], by a significant margin. This underscores the advantages of using point-cloud representations for human clothing models. In comparisons with POP [121], our technique demonstrates superior or competitive performance in both CD and NML across various evaluated outfits. These enhancements are likely due to our use of motion-dependent feature encoding and progressive refinement through a diffusion process. Moreover, our method slightly improves upon the CD and NML scores reported by SkiRT [119] for most outfits. Notably, we achieve a 13.4%

improvement in CD for the challenging long dress outfit ("felice-004"), with scores of 5.58 compared to SkiRT's 6.45. Although SkiRT utilizes a coarse-to-fine strategy, our results affirm the value of multi-step progressive refinement for challenging outfits. Similar observations can be made when contrasting our method with CloSET [244]. While CloSET encodes features on the continuous body surface instead of UV map, it overlooks the importance of dynamics and progressive modeling of clothed humans. The qualitative results depicted in Fig. 7.4 demonstrate that our model generates motion-dependent clothed humans with smooth and natural cloth wrinkles. These results are notably superior to those from the four pose-dependent baselines [111], [121], [174], [244], highlighting the importance of dynamics modeling for this task. Moreover, our method produces denser point-clouds in regions where the clothing is distanced from the human body, an improvement attributable to the progressive modeling facilitated by our diffusion-based framework.

The quantitative results on CAPE dataset are summarized in Tab. 7.2, which includes the results for the "blazerlong" and "shortlong" outfits, as well as the average performance across all outfits. When compared to SCALE [120], our method demonstrates substantial improvements for both listed outfits. The inferiority of SCALE is primarily due to its reliance on low-resolution point-clouds. In relation to POP [121], our technique yields improvements of 8.5% and 1.8% in CD and NML, respectively, when averaged across all outfits. These gains attest to the efficacy of our motion-dependent feature encoding and progressive modeling techniques for clothed human figures. Furthermore, these factors also contribute to our method's outperformance of CloSET [244]. While CloSET does achieve superior results by learning features on a continuous body surface, it falls short in adequately addressing the dynamics and progressive aspects of clothed human modeling.

7.3.2 Ablation Study

In this section, we conduct an ablation study to investigate the contributions of two core components in our approach: motion-dependent temporal feature encoding and the progressive diffusion process.

Methods	Outfits																	
	christine-027			janett-025			felice-004			alexandra-006			eric-035			all		
	knee dress short sleeve			short skirt long sleeve			long dress tank top			loose blouse long pants			blazer jacket long pants			-		
	CD	NML	DE	CD	NML	DE	CD	NML	DE	CD	NML	DE	CD	NML	DE	CD	NML	DE
Ours	1.35	0.97	3.20	1.03	0.81	2.79	5.58	1.14	3.08	1.50	1.29	3.10	1.30	1.15	3.71	1.14	1.01	3.12
w/o temp.	1.39	0.97	3.32	1.10	0.83	2.85	5.87	1.16	3.30	1.56	1.29	3.19	1.31	1.16	3.92	1.20	1.02	3.23
w/o diff.	1.66	0.97	3.25	1.20	0.87	2.82	6.93	1.23	3.16	1.67	1.30	3.12	1.33	1.16	3.78	1.30	1.02	3.18
w/o both	1.72	0.97	3.36	1.23	0.89	2.88	7.30	1.24	3.37	1.70	1.31	3.25	1.35	1.17	3.99	1.36	1.02	3.27

Table 7.3: Quantitative results of ablation study on ReSynth dataset. In each column, we underline key comparisons for enhanced clarity of ablation results.

The study comprises three scenarios: 1) Without dynamics modeling, labeled by *w/o temp.*, means that only pose-dependent features are considered without motion-dependent temporal features. 2) Without progressive modeling, denoted by *w/o diffusion*, indicates that the progressive denoising process is omitted, and predictions are predicted in a single stage only. 3) Without both components, denoted by *w/o both*, refers to the exclusion of both aforementioned elements, resulting in a pipeline similar to that of POP [121].

We present the quantitative results of our ablation study on the ReSynth dataset in Tab. 7.3. For enhanced clarity, we underline key comparisons in each column. Significantly, the model without motion-dependent temporal feature encoding shows substantially larger errors in capturing the dynamics of clothed humans across a variety of outfits. This is particularly evident in the "christine-027", "felice-004" and "eric-035" outfits. These findings emphasize the critical role of motion-dependent features encoding for the *dynamics modeling* of clothed humans. Meanwhile, when the diffusion component is absent, there is a noticeable increase in CD errors, especially in the challenging cases involving loose outfits, such as "felice-004," where the score rises from 5.58 to 6.93. Similar trends are observed in three other outfits: "christine-027", "janett-025" and "alexandra-006". These results underscore the efficacy of our proposed multi-step denoising process, wherein *progressive modeling* significantly enhances the representation of loose-fitting garments relative to the underlying unclothed bodies.

7.4 Conclusion

In this chapter, we introduce ClothDiffuse, an innovative end-to-end diffusion-based method that integrates all three significant aspects, dynamics modeling, progressive modeling, and diversified modeling, for the modeling of clothed humans in a motion. Empirical results show that ClothDiffuse surpasses existing benchmarks, marking a significant advance in the field of clothed human modeling.

Chapter 8

Conclusion

8.1 Summary

In summary, this thesis focuses on the field of human pose estimation and shape modeling from and particularly beyond RGB cameras, which attempts to explore the potential opportunities presented by emerging cameras, such as event cameras, polarization cameras and point-clouds.

Specifically, RGB cameras, renowned for their widespread use and cost-effectiveness, excel in a multitude of applications ranging from everyday photography to basic surveillance systems. However, when it comes to industrial scenarios demanding low power consumption and instantaneous response, such as in sophisticated video surveillance systems or the nuanced object detection required in autonomous vehicles, event cameras emerge as a superior alternative. These cameras, distinguished by their ability to capture pixel-level changes at high speed, are particularly advantageous in dynamic, fast-paced environments. On the other hand, tasks that necessitate detailed depth perception and 3D modeling, like intricate human shape modeling or advanced spatial analysis, reveal the limitations of standard RGB cameras and event streams. In these contexts, polarization imaging and point cloud technologies offer viable solutions. Polarization cameras, which detect light waves' orientation, can discern surface characteristics and angles with precision, making them ideal for complex geometrical modeling. Point clouds, generated by depth sensors, provide comprehensive 3D spatial data but come with higher costs and typically lower frame rates compared to conventional imaging meth-

ods. Each imaging modality, therefore, presents a unique spectrum of strengths and weaknesses. The choice of technology hinges on balancing these attributes against the specific needs and constraints of the intended application. In light of this, our continued research into these advanced camera technologies leads us to advocate for the integration of various sensor types.

Our research in this thesis is structured around three pivotal components: the exploration of new cameras, the development of novel approaches, and the creation of large-scale multi-modality datasets for human pose estimation and shape modeling. Consequently, we end up with the corresponding chapters in this thesis.

Chapter 3 (RGB cameras): Multi-person pose estimation, tracking and motion forecasting from RGB videos are usually more practical in real-world applications, but with more challenging cases due to the intra-frame occlusion of multiple persons. In Chapter 3, we introduce a unified framework to address these interconnected tasks. Our framework leverages spatiotemporal deformable attention to encode the relationships between images, effectively overcoming the issue of intra-frame occlusion frequently encountered in multi-person settings. Unlike existing methods that usually focus on isolated tasks or employ cascaded strategies, our unified approach acknowledges the interdependence of pose estimation and tracking tasks. Accurate 3D pose estimation aids robust tracking, which in turn offers valuable regions-of-interest for further pose estimation and serves as a basis for sensible future motion prediction. Through extensive experiments, we demonstrate that our generic model outperforms specialized baselines, exhibiting competitive performance across all three crucial tasks of pose estimation, tracking, and motion forecasting.

Chapter 4 and 5 (Event Cameras): In these two chapters, we have turned our attention to the energy-efficient event cameras for human parametric pose and shape tracking. These cameras offer unique advantages including high temporal resolution, low latency and low power consumption. In contrast to frame-based cameras, event cameras asynchronously register changes in pixel brightness, resulting in a much sparser and more efficient data stream. Recognizing the untapped potential of event data for 3D human pose and shape

estimation, we have developed innovative approaches that primarily rely on event signals. Initially, we introduced a method using optical flow inferred from events, accompanied by a coherence loss function for consistency between event-based and shape-based flows. To evaluate our work, we created the Multi-Modality Human Pose and Shape Dataset (MMHPSD), the first public dataset of its kind, featuring multiple imaging modalities including event cameras. Subsequently, we pushed the envelope further by developing a novel end-to-end sparse deep learning approach based on Spiking Neural Networks (SNNs), which we refer to as Spatiotemporal Spiking Transformer. This model outperforms existing state-of-the-art methods while requiring only about 20% of the computational resources and 3% of the energy consumption. To support this advanced research, we also constructed a larger dataset, SynEventHPD, dedicated to event-based 3D human pose tracking. This dataset is more than ten times larger than the existing MMHPSD, featuring 45.72 hours of event streams and covering a broad range of motions.

Chapter 6 (Polarization Cameras): In this chapter, we introduce a new imaging modality using polarization cameras to estimate human pose and reconstruct clothed human shapes. Our proposed two-stage approach, HumanSfP, leverages geometric cues from single polarization images to estimate human pose and shape. The first stage, Polar2Normal, predicts surface normal maps by incorporating physical laws as priors. These predictions then guide the second stage, Polar2Shape, in reconstructing a clothed human shape, informed by both the surface normal and an initial parametric shape estimated. To facilitate this research, we have created the Polarization Human Shape and Pose Dataset (PHSPD), comprising approximately 527K frames, 21 subjects, 31 unique actions, and around 9.5 hours of recorded video. Empirical evaluations on synthetic and real-world datasets validate the efficacy of our approach, suggesting that polarization cameras offer a promising alternative to traditional RGB cameras for 3D human pose and shape estimation.

Chapter 7 (Point-clouds): With the advancements in hardware technology, depth sensors have emerged as a straightforward tool to capture 3D point-clouds of objects, known for their intuitive, efficient, and general representa-

tion. Meanwhile, clothed human shapes offer a more comprehensive representation of clothing details. Going beyond static human shape reconstruction, our research focuses on the complex task of animating clothed humans with natural clothing deformations, leveraging point-cloud sequences that provide valuable geometric insights into the structure of the clothing. Traditional techniques, relying either on simplistic rigging-and-skinning or physics-based simulations, require intensive computations and specialized expertise to create a simulation-ready clothing mesh. Recent data-driven methods have made strides, but they still lack in several key areas: dynamics modeling of clothing deformations in a motion sequence, progressive modeling for iterative refinements of clothing deformations, and diversified modeling to capture variations in clothing wrinkle patterns. Our diffusion-based approach integrates these three essential elements and learns the dynamics of clothing deformations for the realistic generation of clothing details in target motion animations.

The comprehensive experimental validations undertaken across these different projects substantiate the effectiveness of our proposed methods. Collectively, this work marks a significant advancement in computer vision, providing robust, efficient, and versatile solutions for human pose estimation and shape modeling.

In addition to these technical contributions, we have developed large-scale, multi-modal datasets—PHSPD, MMHPSD, and SynEventHPD—that serve as valuable resources for the research community. These datasets not only set a new benchmark in terms of scale and diversity but also open avenues for future research by incorporating multiple sensing modalities beyond traditional RGB images.

8.2 Outlook

Human pose estimation and shape modeling are crucial components in computer vision, with applications in an array of sectors, including but not limited to AR/VR, gaming, film, healthcare, and digital humans.

Future research could beneficially explore a wider range of emerging cam-

eras or data modalities in the field of human pose estimation and shape modeling. For example, light field cameras, which capture the intensity of light from various directions in a scene, allow for post-capture focus adjustments and 3D imaging. Additionally, thermal imaging cameras, increasingly more compact and affordable, offer unique insights based on heat signatures and are finding applications in healthcare, building inspections, and wildlife monitoring. Beyond imaging modalities, Inertial Measurement Units (IMUs) are also noteworthy. They measure and report a body's specific force, angular rate, and occasionally orientation, through a combination of accelerometers, gyroscopes, and sometimes magnetometers, with recent studies investigating their use in pose estimation. Ultrasonic sensors, which detect distances and movements using ultrasonic waves, represent another potential avenue for exploration.

In addition to the exploration of emerging cameras or data modalities, there are also pronounced challenges and bottlenecks to explore for future research.

One challenge could be *the integration of multiple modalities and physics-based constraints* into the existing frameworks. The integration of multi-modal data such as inertial measurement unit, accelerometers, gyroscopes, and biometric sensors with the image data could supplement the shortcomings of a single imaging modality, and thus drastically improve the robustness and accuracy of pose and shape analysis. Additionally, integrating physics-based constraints could rectify physically implausible pose predictions, creating a more realistic and reliable model. By leveraging this multi-modal fusion and physical realism, we could build systems that are both comprehensive and deeply rooted in real-world mechanics.

Another challenge might be the ability to handle *challenging poses or motions under various conditions*, such as moving cameras, partial occlusions, and intricate backgrounds. One potential pathway to resolve this issue could be the utilization of unsupervised or semi-supervised learning techniques. Building foundation models trained on large-scale unlabeled data could make the systems better at generalizing across different conditions. These models could subsequently be fine-tuned with smaller, labeled datasets for specific applications, thus creating a more adaptive and resilient pose estimation framework.

Another under-explored yet highly pertinent area is *data acquisition*, particularly for broader applications in digital humans. The digital human paradigm not only demands high-fidelity pose and shape modeling but also requires detailed data on other human aspects like speech, text, motion, facial expression, hand gestures, and hair and clothing simulation. The absence of comprehensive datasets that combine all these aspects has been a major roadblock. However, if a large, rich dataset comprising all these elements could be constructed or synthesized, we would likely see significant advancements in AI-Generated Content (AIGC) specific to digital humans. This could revolutionize the way we interact with digital humans, making them more lifelike and responsive across various communication modes.

In summary, the challenges of integrating multiple modalities and physics-based constraints, dealing with complex and variable conditions, and acquiring comprehensive, multi-faceted data represent some pressing bottlenecks in human pose estimation and shape modeling. Tackling these issues effectively will likely yield transformative results, pushing the boundaries of what is currently achievable in applications ranging from AR/VR to the next generation of digital humans.

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," in *IEEE International Conference on Machine Learning*, 2018.

[2] I. Akhter and M. J. Black, "Pose-conditioned joint angle limits for 3d human pose reconstruction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[3] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, *et al.*, "A low power, fully event-based gesture recognition system," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[4] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele, "PoseTrack: A benchmark for human pose estimation and tracking," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[5] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele, "Posetrack: A benchmark for human pose estimation and tracking," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[6] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

[7] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: Shape completion and animation of people," in *ACM transactions on graphics*, ACM, vol. 24, 2005, pp. 408–416.

[8] G. A. Atkinson and E. R. Hancock, "Recovery of surface orientation from diffuse polarization," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1653–1664, 2006.

[9] Y. Ba, A. Gilbert, F. Wang, J. Yang, R. Chen, Y. Wang, L. Yan, B. Shi, and A. Kadambi, "Deep shape from polarization," in *European Conference on Computer Vision*, 2020.

[10] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker, "Detailed human shape and pose from images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2007.

[11] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," *ACM Transactions on Graphics*, vol. 26, no. 3, 72–es, 2007.

[12] J. Bednarik, S. Parashar, E. Gundogdu, M. Salzmann, and P. Fua, "Shape reconstruction by learning differentiable surface representations," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[13] C. Benedek, B. Gálai, B. Nagy, and Z. Jankó, "Lidar-based gait analysis and activity recognition in a 4d surveillance system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 1, pp. 101–113, 2016.

[14] A. Benzine, F. Chabot, B. Luvison, Q. C. Pham, and C. Achard, "PandaNet: Anchor-based single-shot multi-person 3d pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[15] A. Benzine, B. Luvison, Q. C. Pham, and C. Achard, "Single-shot 3d multi-person pose estimation in complex images," *Pattern Recognition*, vol. 112, p. 107 534, 2021.

[16] H. Bertiche, M. Madadi, E. Tylson, and S. Escalera, "Deepsd: Automatic deep skinning and pose space deformation for 3d garment animation," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[17] B. L. Bhatnagar, C. Sminchisescu, C. Theobalt, and G. Pons-Moll, "Combining implicit function learning and parametric models for 3d human reconstruction," in *European Conference on Computer Vision*, 2020.

[18] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll, "Multi-garment net: Learning to dress 3d people from images," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[19] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it smpl: Automatic estimation of 3d human pose and shape from a single image," in *European Conference on Computer Vision*, 2016.

[20] R. Bollapragada, J. Nocedal, D. Mudigere, H.-J. Shi, and P. T. P. Tang, "A progressive batching l-bfgs method for machine learning," in *IEEE International Conference on Machine Learning*, 2018.

[21] A. Burov, M. Nießner, and J. Thies, "Dynamic surface function networks for clothed human bodies," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[22] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[23] E. Calabrese, G. Taverni, C. Awai Easthope, S. Skriabine, F. Corradi, L. Longinotti, K. Eng, and T. Delbruck, "Dhp19: Dynamic vision sensor 3d human pose dataset," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[24] Z. Cao, G. H. Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021.

[25] Z. Cao, H. Gao, K. Mangalam, Q.-Z. Cai, M. Vo, and J. Malik, "Long-term human motion prediction with scene context," in *European Conference on Computer Vision*, 2020.

[26] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*, 2020.

[27] C.-H. Chen and D. Ramanan, "3d human pose estimation = 2d pose estimation + matching," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[28] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[29] L. Chen, Y. Zheng, A. Subpa-Asa, and I. Sato, "Polarimetric three-view geometry," in *European Conference on Computer Vision*, 2018.

[30] S. Chen and M. Guo, "Live demonstration: Celex-v: A 1m pixel multi-mode event-based sensor," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[31] X. Chen, T. Jiang, J. Song, J. Yang, M. J. Black, A. Geiger, and O. Hilliges, "Gdna: Towards generative detailed neural avatars," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[32] X. Chen, Y. Zheng, M. J. Black, O. Hilliges, and A. Geiger, "Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[33] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[34] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[35] Y. Cheng, B. Wang, B. Yang, and R. T. Tan, "Graph and temporal convolutional networks for 3d multi-person pose estimation in monocular videos," in *AAAI Conference on Artificial Intelligence*, 2021.

[36] Y. Cheng, B. Wang, B. Yang, and R. T. Tan, "Monocular 3d multi-person pose estimation by integrating top-down and bottom-up networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[37] J. Chibane, T. Alldieck, and G. Pons-Moll, "Implicit functions in feature space for 3d shape reconstruction and completion," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[38] J. Chibane, G. Pons-Moll, *et al.*, "Neural unsigned distance fields for implicit function learning," *Annual Conference on Neural Information Processing Systems*, 2020.

[39] H. Ci, C. Wang, X. Ma, and Y. Wang, "Optimizing network structure for 3d human pose estimation," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[40] E. Corona, A. Pumarola, G. Alenya, G. Pons-Moll, and F. Moreno-Noguer, "Smplicit: Topology-aware generative model for clothed people," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[41] Z. Cui, J. Gu, B. Shi, P. Tan, and J. Kautz, "Polarimetric multi-view stereo," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[42] I. M. Daly, M. J. How, J. C. Partridge, S. E. Temple, N. J. Marshall, T. W. Cronin, and N. W. Roberts, "Dynamic polarization vision in mantis shrimps," *Nature communications*, vol. 7, p. 12 140, 2016.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

[44] S. Deng and S. Gu, "Optimal conversion of conventional artificial neural networks to spiking neural networks," in *International Conference on Learning Representation*, 2021.

[45] T. Deprelle, T. Groueix, M. Fisher, V. Kim, B. Russell, and M. Aubry, "Learning elementary structures for 3d shape generation and matching," *Annual Conference on Neural Information Processing Systems*, 2019.

[46] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Annual Conference on Neural Information Processing Systems*, 2021.

[47] E. Dibra, H. Jain, C. Oztireli, R. Ziegler, and M. Gross, "Human shape from silhouettes using generative hks descriptors and cross-modal neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[48] E. Dibra, H. Jain, C. Öztireli, R. Ziegler, and M. Gross, "Hs-nets: Estimating human body shape from silhouettes with convolutional neural networks," in *IEEE International Conference on 3D Vision*, 2016.

[49] Z. Dong, C. Guo, J. Song, X. Chen, A. Geiger, and O. Hilliges, "Pina: Learning a personalized implicit neural avatar from a single rgb-d video sequence," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representation*, 2020.

[51] M. Fabbri, F. Lanzi, S. Calderara, S. Alletto, and R. Cucchiara, "Compressed volumetric heatmaps for multi-person 3d pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[52] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, "Learning to detect and track visible and occluded body joints in a virtual world," in *European Conference on Computer Vision*, 2018.

[53] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[54] H. Fan, Y. Yang, and M. Kankanhalli, "Point 4d transformer networks for spatio-temporal modeling in point cloud videos," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[55] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3d pose estimation," in *AAAI Conference on Artificial Intelligence*, 2018.

[56] W. Fang, Y. Chen, J. Ding, D. Chen, Z. Yu, H. Zhou, Y. Tian, and other contributors, *Spikingjelly*, `https://github.com/fangwei123456/spikingjelly`, Accessed: 2023-02-28, 2020.

[57] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, and Y. Tian, "Deep residual learning in spiking neural networks," *Annual Conference on Neural Information Processing Systems*, 2021.

[58] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, "Incorporating learnable membrane time constant to enhance learning of spiking neural networks," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[59] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2022.

[60] G. Gallego, J. E. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, "Event-based, 6-dof camera tracking from photometric depth maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2402–2412, 2017.

[61] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Video to events: Recycling video datasets for event cameras," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[62] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[63] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "Asynchronous, photometric feature tracking using events and frames," in *European Conference on Computer Vision*, 2018.

[64] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, "Detect-and-track: Efficient pose estimation in videos," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[65] R. Girshick, "Fast r-cnn," in *IEEE/CVF International Conference on Computer Vision*, 2015.

[66] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3d surface generation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[67] R. Gu, G. Wang, Z. Jiang, and J.-N. Hwang, "Multi-person hierarchical 3d pose estimation in natural videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4245–4257, 2019.

[68] E. Gundogdu, V. Constantin, A. Seifoddini, M. Dang, M. Salzmann, and P. Fua, "Garnet: A two-stream network for fast and accurate 3d cloth draping," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[69] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[70] I. Habibie, W. Xu, D. Mehta, G. Pons-Moll, and C. Theobalt, "In the wild human pose estimation using explicit 2d features and intermediate 3d representations," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[71] J. Hagenaars, F. Paredes-Vallés, and G. De Croon, "Self-supervised learning of event-based optical flow with spiking neural networks," *Annual Conference on Neural Information Processing Systems*, 2021.

[72] B. Han, G. Srinivasan, and K. Roy, "Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[73] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[74] T. He, Y. Xu, S. Saito, S. Soatto, and T. Tung, "Arch++: Animation-ready clothed human reconstruction revisited," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[75] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Annual Conference on Neural Information Processing Systems*, 2020.

[76] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 2249–2281, 2022.

[77] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *IEEE International Solid-State Circuits Conference*, 2014.

[78] Z. Huang, Y. Xu, C. Lassner, H. Li, and T. Tung, "Arch: Animatable reconstruction of clothed humans," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[79] J. Hwang, J. Lee, S. Park, and N. Kwak, "Pose estimator and tracker using temporal flow maps for limbs," in *International Joint Conference on Neural Networks*, 2019.

[80] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[81] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, "Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors," *IEEE Transactions on Information Theory*, vol. 59, no. 4, pp. 2082–2102, 2013.

[82] B. Jiang, X. Ren, M. Dou, X. Xue, Y. Fu, and Y. Zhang, "Lord: Local 4d implicit representation for high-fidelity dynamic human modeling," in *European Conference on Computer Vision*, 2022.

[83] B. Jiang, J. Zhang, Y. Hong, J. Luo, L. Liu, and H. Bao, "Bcnet: Learning body and cloth shape from a single image," in *European Conference on Computer Vision*, 2020.

[84] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[85] Z. Jiang, Y. Zhang, D. Zou, J. Ren, J. Lv, and Y. Liu, "Learning event-based motion deblurring," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[86] L. Jin, C. Xu, X. Wang, Y. Xiao, Y. Guo, X. Nie, and J. Zhao, "Single-stage is enough: Multi-person absolute 3d pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[87] H. Joo, T. Simon, M. Cikara, and Y. Sheikh, "Towards social artificial intelligence: Nonverbal social signal prediction in a triadic interaction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[88] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews, *et al.*, "Panoptic studio: A massively multiview system for social interaction capture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 190–204, 2017.

[89] A. Kadambi, V. Taamazyan, B. Shi, and R. Raskar, "Depth sensing using geometrically constrained polarization normals," *International Journal of Computer Vision*, vol. 125, no. 1-3, pp. 34–51, 2017.

[90] A. Kamel, B. Sheng, P. Li, J. Kim, and D. D. Feng, "Hybrid refinement-correction heatmaps for human pose estimation," *IEEE Transactions on Multimedia*, vol. 23, pp. 1330–1342, 2020.

[91] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[92] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3d human dynamics from video," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[93] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3d human dynamics from video," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[94] H. Kim, H. Nam, J. Kim, J. Park, and S. Lee, "Laplacianfusion: Detailed 3d clothed-human body reconstruction," *ACM Transactions on Graphics*, vol. 41, no. 6, pp. 1–14, 2022.

[95] J. Kim, I. Hwang, and Y. M. Kim, "Ev-tta: Test-time adaptation for event-based object recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[96] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representation*, 2015.

[97] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[98] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[99] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3d human pose and shape via model-fitting in the loop," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[100] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, 1955.

[101] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, "Unite the people: Closing the loop between 3d and 2d human representations," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[102] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, "Cifar10-dvs: An event-stream dataset for object classification," *Frontiers in Neuroscience*, vol. 11, p. 309, 2017.

[103] J. Li and M. Wang, "Multi-person pose estimation with accurate heatmap regression and greedy association," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5521–5535, 2022.

[104] S. Li, W. Zhang, and A. B. Chan, "Maximum-margin structured learning with deep networks for 3d human pose estimation," in *IEEE/CVF International Conference on Computer Vision*, 2015.

[105] W. Li, H. Liu, H. Tang, P. Wang, and L. Van Gool, "Mhformer: Multi-hypothesis transformer for 3d human pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[106] Y. Li, S. Deng, X. Dong, R. Gong, and S. Gu, "A free lunch from ann: Towards efficient, accurate spiking neural networks calibration," in *IEEE International Conference on Machine Learning*, 2021.

[107] Y. Li, Y. Guo, S. Zhang, S. Deng, Y. Hai, and S. Gu, "Differentiable spike: Rethinking gradient-descent for training spiking neural networks," *Annual Conference on Neural Information Processing Systems*, 2021.

[108] Z. Li, Z. Zheng, H. Zhang, C. Ji, and Y. Liu, "Avatarcap: Animatable avatar conditioned monocular human volumetric capture," in *European Conference on Computer Vision*, 2022.

[109] C.-H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3d object reconstruction," in *AAAI Conference on Artificial Intelligence*, 2018.

[110] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[111] S. Lin, H. Zhang, Z. Zheng, R. Shao, and Y. Liu, "Learning implicit templates for point-based clothed human modeling," in *European Conference on Computer Vision*, 2022.

[112] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014.

[113] Y. Lin, W. Ding, S. Qiang, L. Deng, and G. Li, "Es-imagenet: A million event-stream classification dataset for spiking neural networks," *Frontiers in Neuroscience*, p. 1546, 2021.

[114] L. Liu, Y. Zheng, D. Tang, Y. Yuan, C. Fan, and K. Zhou, "Neuroskinning: Automatic skin binding for production characters with deep graph networks," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–12, 2019.

[115] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[116] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Transactions on Graphics*, vol. 34, no. 6, 248:1–248:16, 2015.

[117] Y. Lu, J.-H. Cha, S.-K. Youm, and S.-W. Jung, "Parametric shape estimation of human body under wide clothing," *IEEE Transactions on Multimedia*, 2020.

[118] Q. Ma, S. Saito, J. Yang, S. Tang, and M. J. Black, "Scale: Modeling clothed humans with a surface codec of articulated local elements," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[119] Q. Ma, J. Yang, M. J. Black, and S. Tang, "Neural point-based shape modeling of humans in challenging clothing," in *International Conference on 3D Vision*, 2022.

[120] Q. Ma, J. Yang, A. Ranjan, S. Pujades, G. Pons-Moll, S. Tang, and M. J. Black, "Learning to dress 3d people in generative clothing," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[121] Q. Ma, J. Yang, S. Tang, and M. J. Black, "The power of points for modeling humans in clothing," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[122] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[123] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *European Conference on Computer Vision*, 2018.

[124] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *IEEE/CVF International Conference on Computer Vision*, 2017.

[125] A. Martínez-González, M. Villamizar, O. Canévet, and J.-M. Odobez, "Efficient convolutional neural networks for depth-based multi-person pose estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4207–4221, 2020.

[126] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation in the wild using improved cnn supervision," in *IEEE International Conference on 3D Vision*, 2017.

[127] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "Xnect: Real-time multi-person 3d motion capture with a single rgb camera," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 82–1, 2020.

[128] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3d pose estimation from monocular rgb," in *IEEE International Conference on 3D Vision*, 2018.

[129] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[130] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[131] *Microsoft azure kinect*, `https://azure.microsoft.com/en-us/services/kinect-dk/`, Accessed: 2023-05-09.

[132] M. Mihajlovic, Y. Zhang, M. J. Black, and S. Tang, "Leap: Learning articulated occupancy of people," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[133] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European Conference on Computer Vision*, 2020.

[134] B. S. Mitchell, *An introduction to materials engineering and science for chemical and materials engineers*. John Wiley & Sons, 2004.

[135] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[136] T. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[137] G. Moon, J. Y. Chang, and K. M. Lee, "Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[138] 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*, 2020.

[139] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[140] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi, "Efficiently combining positions and normals for precise 3d geometry," *ACM transactions on graphics*, vol. 24, no. 3, pp. 536–543, 2005.

[141] A. Neophytou and A. Hilton, "A layered model of human body and garment deformation," in *International Conference on 3D Vision*, 2014.

[142] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision*, 2016.

[143] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *IEEE International Conference on Machine Learning*, 2021.

[144] B. X. Nie, P. Wei, and S.-C. Zhu, "Monocular 3d human pose estimation by predicting depth on joints," in *IEEE/CVF International Conference on Computer Vision*, 2017.

[145] X. Nie, J. Feng, J. Zhang, and S. Yan, "Single-stage multi-person pose machines," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[146] G. Ning, Z. Zhang, and Z. He, "Knowledge-guided deep fractal neural networks for human pose estimation," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1246–1259, 2017.

[147] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural body fitting: Unifying deep learning and model based human pose and shape estimation," in *IEEE International Conference on 3D Vision*, 2018.

[148] P. Palafox, N. Sarafianos, T. Tung, and A. Dai, "Spams: Structured implicit parametric models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[149] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[150] S. Park, J. Hwang, and N. Kwak, "3d human pose estimation using convolutional neural networks with 2d pose information," in *European Conference on Computer Vision*, 2016.

[151] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Annual Conference on Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., 2019.

[152] C. Patel, Z. Liao, and G. Pons-Moll, "Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[153] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[154] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3d hands, face, and body from a single image," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[155] G. Pavlakos, X. Zhou, and K. Daniilidis, "Ordinal depth supervision for 3d human pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[156] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3d human pose and shape from a single color image," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[157] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou, "Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[158] 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," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[159] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[160] S. Qian, J. Xu, Z. Liu, L. Ma, and S. Gao, "Unif: United neural implicit functions for clothed human reconstruction and animation," in *European Conference on Computer Vision*, 2022.

[161] Y. Raaj, H. Idrees, G. Hidalgo, and Y. Sheikh, "Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[162] J. Rajasegaran, G. Pavlakos, A. Kanazawa, and J. Malik, "Tracking people with 3d representations," in *Annual Conference on Neural Information Processing Systems*, 2021.

[163] J. Rajasegaran, G. Pavlakos, A. Kanazawa, and J. Malik, "Tracking people by predicting 3d appearance, location and pose," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[164] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[165] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3d deep learning with pytorch3d," *arXiv:2007.08501*, 2020.

[166] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, "Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time," *International Journal of Computer Vision*, vol. 126, no. 12, pp. 1394–1414, 2018.

[167] N. D. Reddy, L. Guigues, L. Pishchulin, J. Eledath, and S. G. Narasimhan, "TesseTrack: End-to-end learnable multi-person articulated 3d pose tracking," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[168] G. Rogez, P. Weinzaepfel, and C. Schmid, "Lcr-net++: Multi-person 2d and 3d pose detection in natural images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1146–1161, 2019.

[169] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[170] V. Rudnev, V. Golyanik, J. Wang, H.-P. Seidel, F. Mueller, M. Elgharib, and C. Theobalt, "Eventhands: Real-time neural 3d hand pose estimation from an event stream," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[171] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in Neuroscience*, vol. 11, p. 682, 2017.

[172] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[173] S. Saito, T. Simon, J. Saragih, and H. Joo, "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[174] S. Saito, J. Yang, Q. Ma, and M. J. Black, "Scanimate: Weakly supervised learning of skinned clothed avatar networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[175] I. Santesteban, N. Thuerey, M. A. Otaduy, and D. Casas, "Self-supervised collision handling via generative 3d garment models for virtual try-on," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[176] N. Sarafianos, B. Boteanu, B. Ionescu, and I. Kakadiaris, "3d human pose estimation: A review of the literature and analysis of covariates," *Computer Vision and Image Understanding*, vol. 152, pp. 1–20, 2016.

[177] J. Shan and C. K. Toth, *Topographic laser ranging and scanning: principles and processing*. CRC press, 2018.

[178] R. Shao, H. Zhang, H. Zhang, M. Chen, Y.-P. Cao, T. Yu, and Y. Liu, "Doublefield: Bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[179] R. Shao, Z. Zheng, H. Zhang, J. Sun, and Y. Liu, "Diffustereo: High quality human reconstruction via diffusion-based stereo using sparse cameras," in *European Conference on Computer Vision*, 2022.

[180] Y. Shen, J. Liang, and M. C. Lin, "Gan-based garment generation using sewing pattern images," in *European Conference on Computer Vision*, 2020.

[181] D. Shi, X. Wei, L. Li, Y. Ren, and W. Tan, "End-to-end multi-person pose estimation with transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[182] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, *et al.*, "Efficient human pose estimation from single depth images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2821–2840, 2012.

[183] W. A. Smith, R. Ramamoorthi, and S. Tozza, "Height-from-polarisation with unknown lighting or albedo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 12, pp. 2875–2888, 2018.

[184] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *IEEE International Conference on Machine Learning*, 2015.

[185] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Richly activated graph convolutional network for robust skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1915–1925, 2020.

[186] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *International Conference on Learning Representation*, 2020.

[187] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[188] L. Sun, C. Sakaridis, J. Liang, Q. Jiang, K. Yang, P. Sun, Y. Ye, K. Wang, and L. V. Gool, "Event-based fusion for motion deblurring with cross-modal attention," in *European Conference on Computer Vision*, 2022.

[189] Y. Sun, Q. Bao, W. Liu, Y. Fu, M. J. Black, and T. Mei, "Monocular, one-stage, regression of multiple 3d people," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[190] Z. Sun, N. Messikommer, D. Gehrig, and D. Scaramuzza, "Ess: Learning event-based semantic segmentation from still images," in *European Conference on Computer Vision*, 2022.

[191] Z. Sun, J. Chen, L. Chao, W. Ruan, and M. Mukherjee, "A survey of multiple pedestrian tracking based on tracking-by-detection framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1819–1833, 2021.

[192] S. Tang, F. Tan, K. Cheng, Z. Li, S. Zhu, and P. Tan, "A neural network for detailed human depth estimation from a single image," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[193] Y. Tian, H. Zhang, Y. Liu, and L. Wang, "Recovering 3d human mesh from monocular images: A survey," *arXiv preprint arXiv:2203.01923*, 2022.

[194] G. Tiwari, N. Sarafianos, T. Tung, and G. Pons-Moll, "Neural-gif: Neural generalized implicit functions for animating people in clothing," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[195] L. Tiwari, B. Bhowmick, and S. Sinha, "Gensim: Unsupervised generic garment simulator," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[196] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3d pose estimation from a single image," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[197] H. Tu, C. Wang, and W. Zeng, "VoxelPose: Towards multi-camera 3d human pose estimation in wild environment," in *European Conference on Computer Vision*, 2020.

[198] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid, "Bodynet: Volumetric inference of 3d human body shapes," in *European Conference on Computer Vision*, 2018.

[199] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[200] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.

[201] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *IEEE International Conference on Machine Learning*, 2017.

[202] T. Von Marcard, G. Pons-Moll, and B. Rosenhahn, "Human pose estimation from video and imus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1533–1547, 2016.

[203] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The pose knows: Video forecasting by generating pose futures," in *IEEE/CVF International Conference on Computer Vision*, 2017.

[204] Z. Wan, Z. Li, M. Tian, J. Liu, S. Yi, and H. Li, "Encoder-decoder with multi-level attention for 3d human shape and pose estimation," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[205] B. Wandt and B. Rosenhahn, "Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[206] C. Wang, J. Li, W. Liu, C. Qian, and C. Lu, "Hmor: Hierarchical multi-person ordinal relations for monocular multi-person 3d pose estimation," in *European Conference on Computer Vision*, 2020.

[207] C. Wang, Y. Wang, Z. Lin, A. L. Yuille, and W. Gao, "Robust estimation of 3d human poses from a single image," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

[208] J. Wang, S. Tan, X. Zhen, S. Xu, F. Zheng, Z. He, and L. Shao, "Deep 3d human pose estimation: A review," *Computer Vision and Image Understanding*, vol. 210, p. 103 225, 2021.

[209] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei, "3d human pose machines with self-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1069–1082, 2020.

[210] L. Wang, T.-K. Kim, and K.-J. Yoon, "Eventsr: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[211] M. Wang, J. Tighe, and D. Modolo, "Combining detection and tracking for human pose estimation in videos," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[212] M. Wang, X. Chen, W. Liu, C. Qian, L. Lin, and L. Ma, "Drpose3d: Depth ranking in 3d human pose estimation," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2018.

[213] S. Wang, M. Mihajlovic, Q. Ma, A. Geiger, and S. Tang, "Metaavatar: Learning animatable clothed human models from few depth images," *Annual Conference on Neural Information Processing Systems*, 2021.

[214] T. Wang, J. Zhang, Y. Cai, S. Yan, and J. Feng, "Direct multi-view multi-person 3d human pose estimation," in *Annual Conference on Neural Information Processing Systems*, 2021.

[215] X. Wang, L. Gao, Y. Zhou, J. Song, and M. Wang, "Ktn: Knowledge transfer network for learning multi-person 2d-3d correspondences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7732–7745, 2022.

[216] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[217] Z. Wang, X. Nie, X. Qu, Y. Chen, and S. Liu, "Distribution-aware single-stage models for multi-person 3d pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[218] R. Wehner and M. Müller, "The significance of direct sunlight and polarized skylight in the ant's celestial system of navigation," *Proceedings of the National Academy of Sciences*, vol. 103, no. 33, pp. 12 575–12 579, 2006.

[219] G. Wei, C. Lan, W. Zeng, and Z. Chen, "View invariant 3d human pose estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4601–4610, 2019.

[220] W.-L. Wei, J.-C. Lin, T.-L. Liu, and H.-Y. M. Liao, "Capturing humans in motion: Temporal-attentive 3d human pose and shape estimation from monocular video," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[221] D. Xiang, F. Prada, T. Bagautdinov, W. Xu, Y. Dong, H. Wen, J. Hodgins, and C. Wu, "Modeling clothing as a separate layer for an animatable human avatar," *ACM Transactions on Graphics*, vol. 40, no. 6, pp. 1–15, 2021.

[222] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *European Conference on Computer Vision*, 2018.

[223] M. Xiao, Q. Meng, Z. Zhang, D. He, and Z. Lin, "Online training through time for spiking neural networks," *Annual Conference on Neural Information Processing Systems*, 2022.

[224] Y. Xiu, J. Yang, D. Tzionas, and M. J. Black, "Icon: Implicit clothed humans obtained from normals," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[225] C. Xu, L. N. Govindarajan, Y. Zhang, and L. Cheng, "Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups," *International Journal of Computer Vision*, vol. 123, no. 3, pp. 454–478, 2017.

[226] D. Xu, Y. Jiang, P. Wang, Z. Fan, Y. Wang, and Z. Wang, "Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[227] L. Xu, W. Xu, V. Golyanik, M. Habermann, L. Fang, and C. Theobalt, "Eventcap: Monocular 3d capture of high-speed human motions using an event camera," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[228] Y. Xu, S.-C. Zhu, and T. Tung, "Denserac: Joint 3d pose and shape estimation by dense render-and-compare," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[229] Z. Yan, J. Zhou, and W.-F. Wong, "Near lossless transfer learning for spiking neural networks," in *AAAI Conference on Artificial Intelligence*, 2021.

[230] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, "Tubedetr: Spatio-temporal video grounding with transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[231] L. Yang, F. Tan, A. Li, Z. Cui, Y. Furukawa, and P. Tan, "Polarimetric dense monocular slam," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[232] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, "3d human pose estimation in the wild by adversarial learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[233] Z. Yang, S. Wang, S. Manivasagam, Z. Huang, W.-C. Ma, X. Yan, E. Yumer, and R. Urtasun, "S3: Neural shape, skeleton, and skinning fields for 3d human modeling," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[234] M. Yao, H. Gao, G. Zhao, D. Wang, Y. Lin, Z. Yang, and G. Li, "Temporal-wise attention spiking neural networks for event streams classification," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[235] M. Yao, G. Zhao, H. Zhang, Y. Hu, L. Deng, Y. Tian, B. Xu, and G. Li, "Attention spiking neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[236] X. Yao, F. Li, Z. Mo, and J. Cheng, "Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks," in *Annual Conference on Neural Information Processing Systems*, 2022.

[237] X. Yi, C. Caramanis, and E. Price, "Binary embedding: Fundamental limits and fast algorithm," in *IEEE International Conference on Machine Learning*, 2015.

[238] D. Yu, K. Su, J. Sun, and C. Wang, "Multi-person pose estimation for pose tracking with enhanced cascaded pyramid network," in *European Conference on Computer Vision Workshop*, 2018.

[239] Y. Yuan, U. Iqbal, P. Molchanov, K. Kitani, and J. Kautz, "Glamr: Global occlusion-aware human mesh recovery with dynamic cameras," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[240] A. Zanfir, E. Marinoiu, and C. Sminchisescu, "Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[241] A. Zanfir, E. Marinoiu, M. Zanfir, A.-I. Popa, and C. Sminchisescu, "Deep network for the integrated 3d sensing of multiple people in natural images," in *Annual Conference on Neural Information Processing Systems*, 2018.

[242] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "Motr: End-to-end multiple-object tracking with transformer," in *European Conference on Computer Vision*, 2022.

[243] X. Zeng, A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, and K. Kreis, "Lion: Latent point diffusion models for 3d shape generation," in *Annual Conference on Neural Information Processing Systems*, 2022.

[244] H. Zhang, S. Lin, R. Shao, Y. Zhang, Z. Zheng, H. Huang, Y. Guo, and Y. Liu, "Closet: Modeling clothed humans on continuous surface with explicit template decomposition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[245] J. Zhang, B. Dong, H. Zhang, J. Ding, F. Heide, B. Yin, and X. Yang, "Spiking transformers for event-based single object tracking," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[246] J. Zhang, X. Yang, Y. Fu, X. Wei, B. Yin, and B. Dong, "Object tracking by jointly exploiting frame and event domain," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[247] J. Zhang, L. Tang, Z. Yu, J. Lu, and T. Huang, "Spike transformer: Monocular depth estimation for spiking camera," in *European Conference on Computer Vision*, 2022.

[248] K. Zhang, K. Che, J. Zhang, J. Cheng, Z. Zhang, Q. Guo, and L. Leng, "Discrete time convolution for fast event-based stereo," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[249] Q. Zhang, Y. Jiang, Q. Zhou, Y. Zhao, Y. Liu, H. Lu, and X.-S. Hua, "Single person dense pose estimation via geometric equivariance consistency," *IEEE Transactions on Multimedia*, vol. 25, pp. 572–583, 2023.

[250] S. Zhang, Q. Ma, Y. Zhang, Z. Qian, T. Kwon, M. Pollefeys, F. Bogo, and S. Tang, "Egobody: Human body shape and motion of interacting people from head-mounted devices," in *European Conference on Computer Vision*, 2022.

[251] Y. Zhang, C. Wang, X. Wang, W. Liu, and W. Zeng, "Voxeltrack: Multi-person 3d human pose estimation and tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[252] R. Zhao, Y. Wang, and A. M. Martinez, "A simple, fast and highly-accurate algorithm to recover 3d shape from 2d landmarks on a single image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 3059–3066, 2017.

[253] T. Zhao, S. Li, K. N. Ngan, and F. Wu, "3-d reconstruction of human body shape from a single commodity depth camera," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 114–123, 2018.

[254] J. Zhen, Q. Fang, J. Sun, W. Liu, W. Jiang, H. Bao, and X. Zhou, "Smap: Single-shot multi-person absolute 3d pose estimation," in *European Conference on Computer Vision*, 2020.

[255] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3d human pose estimation with spatial and temporal transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[256] Z. Zheng, H. Huang, T. Yu, H. Zhang, Y. Guo, and Y. Liu, "Structured local radiance fields for human avatar modeling," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[257] Z. Zheng, T. Yu, Y. Liu, and Q. Dai, "Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3170–3184, 2021.

[258] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu, "Deephuman: 3d human reconstruction from a single image," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[259] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3d human pose estimation from monocular video," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[260] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Towards 3d human pose estimation in the wild: A weakly-supervised approach," in *IEEE/CVF International Conference on Computer Vision*, 2017.

[261] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[262] Z. Zhou, Y. Zhu, C. He, Y. Wang, S. Yan, Y. Tian, and L. Yuan, "Spikformer: When spiking neural network meets transformer," in *International Conference on Learning Representation*, 2022.

[263] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," in *Proceedings of Robotics: Science and Systems*, 2018.

[264] H. Zhu, X. Zuo, H. Yang, S. Wang, X. Cao, and R. Yang, "Detailed avatar recovery from single image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[265] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[266] S. Zou, C. Guo, X. Zuo, S. Wang, P. Wang, X. Hu, S. Chen, M. Gong, and L. Cheng, "Eventhpe: Event-based 3d human pose and shape estimation," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[267] S. Zou, Y. Mu, X. Zuo, S. Wang, and L. Cheng, "Event-based human pose tracking by spiking spatiotemporal transformer," *arXiv preprint arXiv:2303.09681*, 2023.

[268] S. Zou, Y. Xu, C. Li, L. Ma, L. Cheng, and M. Vo, "Snipper: A spatiotemporal transformer for simultaneous multi-person 3d pose estimation tracking and forecasting on a video snippet," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[269] S. Zou, X. Zuo, Y. Qian, S. Wang, C. Xu, M. Gong, and L. Cheng, "3d human shape reconstruction from a polarization image," in *European Conference on Computer Vision*, 2020.

[270] S. Zou, X. Zuo, S. Wang, Y. Qian, C. Guo, and L. Cheng, "Human pose and shape estimation from single polarization images," *IEEE Transactions on Multimedia*, 2022.

[271] X. Zuo, S. Wang, J. Zheng, W. Yu, M. Gong, R. Yang, and L. Cheng, "Sparsefusion: Dynamic human avatar modeling from sparse rgbd images," *IEEE Transactions on Multimedia*, vol. 23, pp. 1617–1629, 2020.