Robust 3D Human Modeling for Baseball Sports Analytics

by

Jerrin Bright

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contribution

The thesis includes contributions that were drafted for publication during the writing period. As the lead author, I played a key role in the conceptualization, developing code, and drafting and submitting the manuscripts for publication. Additionally, I also co-authored research papers within the broader field.

Primary Contributions

Jerrin Bright, Bavesh Balaji, Harish Prakash, Yuhao Chen, David Clausi and John Zelek. 2024. Distribution and Depth-Aware Transformers for 3D Human Mesh Recovery. In the 21st Conference on Robots and Vision (CRV'24)

Jerrin Bright, Bavesh Balaji, Yuhao Chen, David Clausi, and John Zelek. 2024. PitcherNet: Powering the Moneyball Evolution in Baseball Video Analytics. In the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'24)

Jerrin Bright, Yuhao Chen and John Zelek. 2023. Mitigating Motion Blur for Robust 3D Baseball Player Pose Modeling for Pitch Analysis. In Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports (MMSports '23).

Secondary Contributions

Bavesh Balaji, **Jerrin Bright**, Sirisha Rambhatla, Yuhao Chen, Alexander Wong, John Zelek and David A Clausi. 2024. Domain-Guided Masked Autoencoders for Unique Player Identification. In the 21st Conference on Robots and Vision (CRV'24)

Bavesh Balaji*, **Jerrin Bright***, Harish Prakash, Yuhao Chen, David A. Clausi, and John Zelek. 2023. Jersey Number Recognition using Keyframe Identification from Low-Resolution Broadcast Videos. In Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports (MMSports '23).

Abstract

In the high-stakes world of baseball, every nuance of a pitcher's mechanics holds the key to maximizing performance and minimizing runs. Traditional analysis methods, reliant on pre-recorded offline numerical data, struggle in the dynamic flow of live games. Though seemingly ideal, broadcast video analysis faces significant challenges due to motion blur, occlusion, and low resolution. This thesis proposes a novel 3D human modeling technique and a pitch statistics identification system that are robust to the aforementioned challenges.

Specifically, we propose D2A-HMR, a depth and distribution-aware 3D human mesh recovery technique that extracts pseudo-depth from each frame and utilizes a transformer network with self- and cross-attention to create a 3D mesh that extracts the 3D pose coordinates. The network is regularized using various loss functions including a silhouette loss function, joint reprojection loss functions, and a distribution loss function which utilize normalizing flow to learn the deviation between the underlying predicted and ground truth distributions. Furthermore, we propose a focused augmentation strategy specifically designed to address the motion blur issue caused by fast-moving motion.

Following that, we introduce the PitcherNet system, which is built upon the D2A-HMR and motion blur augmentation strategy. PitcherNet proposes an automated analysis system that analyzes pitcher kinematics directly from live broadcast video, providing valuable pitch statistics (pitch velocity, release point, pitch position, release extension, and pitch handedness). The system relies solely on the broadcast videos as its input and leverages computer vision and pattern recognition to generate reliable pitch statistics from the game. Firstly, PitcherNet isolates the 'pitcher' and 'batter' from each frame. A novel role classification network by decoupling actions from pitcher kinematics is utilized to accomplish this task. Next, PitcherNet generates 18 keypoints representing the pitcher's joints using a refined version of D2A-HMR model.

Additionally, we enhance the generalizability of the 3D human model by incorporating additional in-the-wild high-resolution videos from the Internet. Finally, PitcherNet employs Temporal Convolutional Networks (TCN) and kinematic-driven heuristics to capture the pitch statistics which can be used to analyze baseball pitchers.

Acknowledgements

I would like to express my sincere gratitude to all those who have supported me throughout my graduate journey and made this thesis possible.

First and foremost, I am incredibly grateful to my advisor, Dr. John Zelek, for his invaluable guidance and unwavering support. His insightful feedback, encouragement, and expertise have been instrumental in shaping this thesis and propelling my research into exciting new directions. I am truly fortunate to have benefited from his mentorship.

I also would like to extend my appreciation to the Baltimore Orioles of Major League Baseball, specifically Sig Mejdal, Di Zou, and James Hull. Their collaboration and provision of crucial resources have been instrumental in the success of this thesis. Additionally, I am grateful to MITACS for their financial support through the Mitacs Accelerate program.

A special thank you to Dr. Yuhao Chen and Dr. David Clausi for their technical support throughout my research endeavors. Their assistance was invaluable in overcoming challenges and ensuring the smooth progress throughout my journey.

Finally, I want to express my heartfelt gratitude to my family and friends. To my parents, Edwin and Felci, thank you for everything you have done for me. You have instilled in me the values that have allowed me to reach this point. Thank you to my brothers Derrin and Darrin. Your presence in my life means the world to me.

Dedication

This work is dedicated to my parents for their sacrifices, support, and belief in me.

Table of Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

SMPL Skinned Multi-Person Linear

HMR Human Mesh Recovery

CNN Convolutional Neural Networks

LHC Local Histogram Correlation

GHC Global Histogram Correlation

RoI Region of Interest

MAE Masked Autoencoders

KfID Keyframe Identification

ViT Vision Transformer

DTW Dynamic Time Warping

OOD Out-Of-Distribution

MLP Multilayer Perceptron

PA-mPJPE Procrustes-Aligned mean Per Joint Position Error

mPJPE mean Per Joint Position Error

mPVE mean Per Vertex Error

COCO Common Objects in COntext

RAFT Recurrent All-Pairs Field Transforms

SOTA State-Of-The-Art

LR Learning Rate

ItW In-the-Wild

SABR Society of American Baseball Research

SVM Support Vector Machines

LDA Linear Discriminant Analysis

TCN Temporal Convolutional Network

3DPW 3D Poses in the Wild

GANs Generative Adversarial Networks

MANO hand Model with Articulated and Non-rigid defOrmations

Chapter 1

Introduction

In recent years, the advent of deep learning has revolutionized various fields, enabling remarkable performance improvements. This phenomenon has now extended its influence into the realm of sports analytics, particularly in major team sports such as baseball [107, 16, 41], ice-hockey [33, 116, 115], basketball [22, 110, 78] and soccer [37, 7, 38] which enjoy extensive global viewership and participation. Teams across these sports are increasingly turning to vision-driven analytics to gain a competitive edge by evaluating player performance and making informed assessments.

Sabermetrics, pioneered by the Society of American Baseball Research (SABR) [34], the empirical analytics approach to in-game baseball analysis, has seen remarkable growth in recent times [60]. Although most of Sabermetrics work focuses on structured statistical data [107, 41] such as utilizing offline data like pitch type, break, spin rate, and historical win rate; video analysis offers the potential for visual understanding, detailed performance evaluation, and contextual information from real-time data. Baseball, often regarded as a sport with extensive statistical analysis, provides valuable insights into various aspects of the game and player skills [51, 50]. Being a pitcher-friendly game, the performance of the pitcher significantly influences the team's success and overall gameplay. Analyzing pitchers in baseball is of utmost importance, as it can significantly enhance the assessment of pitching techniques, accurately evaluate pitch movements, and aid in detecting subtle patterns, such as changes in delivery or pitch tipping. This comprehensive analysis not only provides valuable information about individual pitchers, but also significantly contributes to improving the overall performance of the team.

1.1 Problem Description

Analyzing a pitcher's mechanics in real-time from live broadcasts presents several technical challenges. Unlike the controlled environments used in traditional motion capture [84, 92, 103], broadcast videos introduce inherent difficulties due to the dynamic nature of the game.

Motion blur presents a significant challenge. During the pitching motion, a pitcher's arm moves extremely fast, often exceeding 100 mph. At typical frame rates like 30fps, this rapid movement can result in blurry frames. This blurring is particularly problematic for fast pitches. Unfortunately, blurry frames make it difficult for traditional computer vision algorithms to pinpoint the exact location of the pitcher's joints. This limitation hinders the ability to track joint movement and accurately reconstruct a 3D representation of the pitcher's pose.

Another challenge is *occlusion*. This occurs when other players on the field, objects like bats or balls, or even the camera angle itself, block the view of the pitcher's body. These blockages, also known as self-occlusions, can lead to missing data about the pitcher's pose. This incomplete data can then negatively impact the accuracy of both the 3D human modeling process and the subsequent extraction of pitch mechanics data.

The *lower resolution* of broadcast videos creates another hurdle. Compared to high-quality recordings used in controlled environments, broadcast videos often have fewer pixels. For instance, a typical broadcast might be in 720p (1280x720 pixels), whereas a high-quality recording used for analysis might be in 4K (3840x2160 pixels). This reduction in detail can make it challenging to distinguish subtle changes in the pitcher's body position such as the angle of the elbow during wrist cocking. These subtle changes might be crucial for accurately analyzing the mechanics of a pitch.

These technical limitations associated with live broadcast video hinder the ability of existing analysis methods to capture the dynamic nature of a pitcher's mechanics in real-time. Thus, the thesis is driven to address the above-mentioned challenges enabling robust 3D human modeling and subsequently reliable pitch analysis for baseball sport.

1.2 Motivation

Baseball is a game of precision and power, where the mechanics of a pitcher significantly impact their performance and injury risk. However, current analysis methods fail to capture the dynamic nature of live games. Traditional analysis methods using prerecorded data lack

real-time capabilities crucial for in-game adjustments. Although broadcast video analysis seems ideal, technical limitations highlighted in the previous section hinder accurate data extraction.

This gap between existing methods and the need for real-time analysis during live games motivates the development solutions to solve these issues. The proposed solutions in this thesis aims to bridge this gap by leveraging computer vision and pattern recognition directly from live broadcasts. These innovations have the potential to:

1. **Automate Scouting and Evaluation:** Extracted kinematic data can be integrated into player scouting reports, offering a more objective and data-driven evaluation of pitching talent.
2. **Deeper Fan Engagement:** Automated real-time pitch statistics can enrich broadcasts for viewers using just a single smart phone, improving the viewing experience and offering new insights for fans.
3. **Inform Coaching Decisions:** Coaches gain valuable kinematic insights for strategic adjustments and introduce tailored training programs to address specific mechanical deficiencies of the pitchers.
4. **Reduce Injury Risk:** By analyzing subtle changes in mechanics that could indicate potential overuse or stress, injuries can be predicted and prevented.
5. **Automate Referee Assistance:** Extracting the pitch statistics from live video can potentially help umpires make close calls on balls and strikes.

1.3 Major Contributions

The thesis dissertation presents unique contributions that prove the efficacy of 3D human modeling in baseball sports analytics. The primary contributions of the dissertation include the following.

1. We introduce **PitcherNet**, a novel automated system, which enables accurate prediction of baseball pitch statistics from low-quality broadcast videos.
2. We introduce a novel image-based HMR model named **D2A-HMR** that adeptly models the underlying distributions and integrates pseudo-depth priors for efficient and accurate mesh recovery.

3. Taking advantage of the **residual log-likelihood** approach, we refine the 3D human model by learning the disparity between the predicted underlying distribution and the ground truth.

4. We propose a **focused augmentation strategy** that incorporates motion blur artifacts, challenging the conventional belief in complex pipelines and showing significant improvements in handling these challenges.

5. We propose an **innovative pitcher identification strategy** which aims at player role classification by decoupling actions from player kinematics.

The secondary contributions of the dissertation include:

1. We propose a new domain-guided masking strategy, termed ***d*-MAE**, specifically tailored to player identification, enhancing model robustness to motion blur.

2. We introduce a pipeline that incorporates **in-the-wild data** from the Internet, capturing the variability and complexity present in the data, resulting in an efficient and versatile pose estimation.

3. We propose a **keyframe identification module** that is robust to blur and occlusions using Region of Interest (RoI) and Spatial Context-Aware filtering to facilitate effective jersey number recognition.

1.4 Thesis Outline

The thesis is structured to address challenges in kinematic-driven pitch analysis for baseball videos due to agile actions and limitations in datasets. Chapter 2 provides background on different human datasets, evaluation metrics, 3D human modeling, and existing pose estimation and action recognition approaches. Chapter 3 details the baseball video dataset and its preprocessing. Chapters 4 and 5 discuss the proposed algorithms for 3D pose modeling during agile actions and how they address motion blur. Chapter 6 explains how the generated 3D models are used to derive kinematically-driven pitch statistics. Finally, Chapter 7 concludes the thesis by summarizing the findings, contributions, impact, and potential applications of the research.

Chapter 2

Background

This chapter discusses the relevant background for tmetrics,arch presented in Chapters 4, 5 and 6. An overview of human pose estimation datasets is provided first, placing emphasis on the datasets used for training and validating the models developed in this thesis. An extensive overview on human modeling and human pose estimation is then given. This is followed by some literature discussing the action recognition techniques adapted for sports.

2.1 Datasets

In this thesis, we primarily focussed on one in-house dataset (MLBPitchDB) and two publically released datasets containing annotations for human joint positions. Each of these datasets are explained in more detail in the following sections.

MLBPitchDB. The MLBPitchDB dataset [16] is specifically built for effective baseball sports analysis. It offers comprehensive data encompassing player details, 3D pose estimations, player actions, and detailed play statistics for all players within the camera's view. These statistics include pitch extension, velocity, release point, and various spin characteristics. To ensure data quality, the dataset undergoes preprocessing techniques outlined in [16], including player detection, data synchronization, and camera re-projection. Further details on the dataset and preprocessing methods can be found in Chapter 3.

Human3.6M. Human3.6M dataset [44] is the largest publicly available collection for 3D human pose estimation tasks. This dataset features 3.6 million images showcasing 7 profes-

sional actors engaged in 15 common activities, including walking, eating, and conversation. It provides both 2D and 3D ground truth poses for each image, along with camera calibration parameters and individual body measurements for the actors. Following established practice within the field [70, 49], our training process utilized data from subjects S1, S5, S6, S7, and S8, while evaluations were conducted on subjects S9 and S11.

3DPW. The 3D Poses in the Wild (3DPW) [118] dataset offers a unique resource for evaluating 3D human pose estimation in natural environments. Unlike prior datasets captured in controlled settings, 3DPW leverages videos recorded with a moving phone camera, showcasing real-world scenarios. It comprises 60 video sequences featuring diverse human actions. Each sequence is richly annotated with 2D and 3D pose information for every frame, along with camera data and 3D body scans of the individuals involved. For training and testing purposes, the dataset adheres to the standard split of 22,000 and 35,000 images, respectively.

2.2 Evaluation Metrics

We employ several metrics to assess the performance of different components of our system. For the 3D human pose estimation task, we utilize two primary metrics: the mean Per Joint Position Error (mPJPE) and the Procrustes-Aligned mean Per Joint Position Error (PA-mPJPE). These metrics quantify the average distance between the predicted and ground truth 3D joint locations, with PA-mPJPE accounting for global pose variations. For player tracking and identification, we evaluate the system's accuracy performance.

In line with established practices from previous research [54, 70, 28], we subjected our 3D human model to a comprehensive evaluation using key metrics: mPJPE, PA-mPJPE and mean Per Vertex Error (mPVE) in both the 3DPW and Human3.6M datasets. mPVE metric is ignored if the ground truth mesh is not available. All metrics were measured in millimeters (mm), providing a precise assessment of our model's performance. Finally, pitch statistics performance is assessed using standard metrics such as F1-score, precision, or accuracy with different classification margins.

2.3 3D Human Modeling

Human Mesh Recovery (HMR) is an approach to estimate the pose and shape of a human subject, featuring a broad spectrum of applications across various downstream tasks [124,

96, 129]. The modeling of the human mesh can significantly improve player performance in the realm of sports. For example, the representation of the player using HMR enables precise biomechanical analysis where the understanding of joint movements, muscle actions, and overall body kinematics can be analyzed. Similarly, potential stress points on the body during actions can be analyzed to prevent injuries. Additionally, data-driven coaching and simultaneous unbiased quantitative performance evaluations of players can be performed to analyze the overall impact of players in the game.

Early attempts to reconstruct the surface of the human body were made by [99, 4, 106, 56], but the introduction of the Skinned Multi-Person Linear (SMPL) [80] revolutionized surface modeling by providing a highly realistic representation capable of capturing deformations. SMPL is detailed extensively in Section 2.3.1 and the types and advances in HMR is detailed in Section 2.3.2.

2.3.1 SMPL

SMPL [79] is a statistical model with the objective of efficiently mapping human subjects with two parameters: shape and pose parameters. The shape parameters β correspond to the 10 shape coefficients of the PCA shape space, where each dimension interprets a different aspect such as height (tall/short) or expansion/shrink in a particular direction or weight of the human subject. The pose parameters θ corresponds to the 23×3 relative 3D joint rotation of 24 joint positions from a parent joint in the human body and ϕ corresponds to the statistical prior of the human body. The SMPL model $M(\theta, \beta) \in \Re^{3\times 6980}$ represents the output of the triangulated mesh obtained from the shaping of the template mesh conditioned on θ and β.

The process of synthesizing humans from the SMPL model using a template mesh can be categorized into Shape Blend Shapes, Pose Blend Shapes, and Skinning. This is qualitatively demonstrated in Figure 2.1.

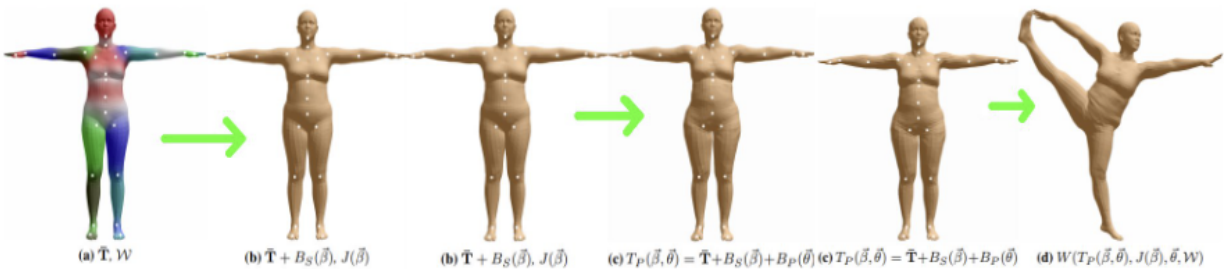


Figure 2.1: Overview of the SMPL process [79].

Shape Blend Shapes. The vertex displacements are added to the template mesh (T) in this step to represent how far the subject shape is from the template shape.

Pose Blend Shapes. The relative rotation of the joints causes deformation on the vertices surrounding the corresponding joints. Thus, the vertex displacements are added which represent the deformation as a result of a specific pose.

Skinning. A weighted combination of the joint deformation is used to transform each vertex of the template mesh to align with the expected pose.

2.3.2 Types of HMR

HMR can be split into two types: parametric and non-parametric approaches. The parametric approach involves the modeling of a network to regress the parameters of a parametric body model from the input image, which are subsequently utilized for human mesh generation, as elucidated in [39, 25]. These methods typically employ Convolutional Neural Networks (CNN) to extract image features and then predict the model parameters from the features. Parametric modeling approaches can further be split into optimization-based and learning-based approaches. Optimization-based approaches fit a body model by minimizing the error between different prior terms. SMPLify [14] fits the parametric SMPL [79] model to minimize the error between the recovered mesh and keypoints. In addition, prior terms including silhouettes [25, 124] or distance functions [134] are used to penalize unrealistic shapes and poses. Learning-based approaches take advantage of deep neural networks to predict model parameters [55, 25, 24]. Recent works including HMR-ViT [24] use a transformer-only temporal architecture to predict the model parameters, and ImpHMR [25] uses neural feature fields to model humans in 3D space from a single image. However, the use of pose parameters as a regression target can introduce inaccuracies and non-minimal representations, leading to performance limitations [56].

Thus, recent works have been witnessed in non-parametric approaches [28, 70, 57], which directly regresses the 3D coordinates of the human mesh without relying on a predefined parametric model. Non-parametric modeling approaches including GraphCMR [58], Pixel2mesh [119], and Feastnet [117] use graphical neural networks to regress vertices from RGB images, as they are effective in modeling neighborhood vertex-vertex interactions. Pose2Mesh [28] uses a 2D and 3D pose to regress the vertices using spectral graphical neural networks. METRO [70] uses transformers to model the global interaction between vertices, and I2LMeshNet [87] uses a heatmap-based representation called lixel to regress the human mesh.

2.4 Pose Estimation

Human pose estimation is one of the fundamental problems in computer vision. Estimating the pose or the joint positions of the humans with a single camera is a very challenging task. It has numerous applications in sports, action recognition, computer-assisted living, human-computer interfaces, special effects, and telepresence [74, 104, 75, 127, 29, 18].

Pose estimation can be divided into 2D and 3D pose estimation approaches and can further be sub-categorized based on the idea behind the approach. In this section, different types of pose estimation are detailed on the basis of their taxonomy.

Determining the 2D pose of the person, that is, the joint positions (X, Y coordinates) of an image is termed the 2D pose estimation [125, 111, 85, 93, 66, 90, 120]. 2D human pose estimation can be done in two ways: top-down and bottom-up approaches. The top-down approach prioritizes and isolates the person and then leverages a model to predict the joint positions. Whereas, bottom-up approach directly scans the image for all the plausible keypoints and then connects the keypoints to form a complete human pose.

Several techniques have been proposed for 2D pose estimation, which can be broadly categorized into heatmap-based and regression-based approaches. Heatmap-based methods [125, 66, 111] focus on predicting heatmaps that represent the likelihood that each keypoint is present at different locations in the image. These heatmaps are then processed to estimate the exact keypoint locations. On the other hand, regression-based methods [93, 85] directly regress the coordinates of keypoints from the input image by employing deep neural networks to learn the mapping between the image and the keypoints. These approaches have demonstrated impressive performance in capturing fine-grained details and handling occlusions, making them suitable for challenging pose estimation tasks.

Determining the 3D pose of the person, that's the joint positions (X, Y, and Z co-ordinates) from an image is termed 3D pose estimation [68, 62, 77, 114, 67, 19]. There are various ways in which the 3D Pose Estimation problem can be approached and some approaches include lifting 2D to 3D, training 3D alongside 2D pose, training 3d directly from images, etc. Recently, transformer-based networks have emerged as State-Of-The-Art (SOTA) models. Epipolar Transformers [40] utilizes epipolar constraints to enforce geometric consistency between 2D keypoints. TransFuse [82] incorporates a cross-modal transformer to fuse information from multiple views. MHFormer [68] introduces multihead self-attention mechanisms to capture both local and global dependencies.

2.5 Player Action Recognition

Deep learning has emerged as a powerful tool for action recognition, offering promising results. The use of 3D convolutions has demonstrated effectiveness in capturing crucial spatio-temporal information from video data [61, 29, 128, 18]. However, these methods often suffer from a large number of parameters, making them susceptible to overfitting in smaller datasets. To address this limitation, Li *et al.* [61] introduced a spatio-temporal attention network, enabling identification of the key video frames and spatially focus on those frames. Similarly, works including [18, 29, 128, 127] leveraged the pose features from each frame of the sequence and enable effective action recognition without introducing parameter overhead.

Yao *et al.* [127] coupled pose and action by formulating pose as an optimization on a set of action-specific manifolds. Cai *et al.* [18] use a two-stage architecture that extracts pose information and temporal information using optical flow technique before combining them. STAR-Transformer [2] fused video and skeletal data using a transformer architecture with a special cross-attention mechanism. SVFormer [123] introduced a semi-supervised learning approach by incorporating a novel data augmentation technique called Tube TokenMix, specifically designed to improve video understanding.

2.6 Summary

This chapter establishes the foundation for 3D human modeling and analysis by introducing critical concepts like pose estimation and action recognition. The chapter reviews commonly used datasets and evaluation metrics for human pose estimation. It then delves into the details of human modeling with SMPL models, exploring the parametric and non-parametric variations. Finally, the chapter surveys recent advancements in 2D and 3D pose estimation methods, along with their underlying paradigms. Subsequent chapters will expand upon these datasets and propose novel methods to address inherent visual challenges within them, with the ultimate goal of achieving robust and reliable 3D human modeling and analysis.

Chapter 3

MLBPitchDB Dataset

This chapter introduces the MLBPitchDB dataset, a collection of baseball pitching data specifically designed to evaluate a network's ability to generalize to unseen data. The dataset contains information from over 1,000 games, encompassing more than 100,000 pitches. It includes data points often missing in standard datasets, such as those featuring unusual poses and extreme motion blur. This makes the MLBPitchDB dataset a valuable tool for researchers developing pose estimation and action recognition algorithms.

Each data point within MLBPitchDB captures 18 3D joint positions of the pitcher, along with detailed pitch statistics. These statistics include pitching velocity, extension, pitch type, horizontal/vertical break, spin rate, set position, release velocity, and other observed pitch metrics. Additionally, the dataset provides the 3D joint positions of both the batter and the catcher, facilitating a more comprehensive analysis of game dynamics.

3.1 Dataset Description

We have used only 150 pitch sequences, which correspond to 30,000 frames in total, to strike a balance between having enough data to train a reliable model and avoiding the risk of overfitting by focusing on a smaller, but more diverse, and representative dataset. The dataset has been divided into three subsets: training, validation, and testing with the respective split provided in Table 3.1.

Table 3.1: Dataset split for Training, Validation, and Testing

Dataset	Pitch Sequences	Frames
Train	105	21,050
Validation	15	2,962
Test	30	5,988

The size and composition of the dataset were carefully chosen after considering the diversity and coverage of pitching actions, including the pitcher handedness, set position, pitch type, and game lighting conditions. Experimentation was conducted exclusively on the test set that contains real-world input frames with inherent motion blur effects.

Some problems with the dataset include extra frames before/after the pitching action for which the pitcher's pose is not annotated and the absence of the camera parameters. Camera parameters are required to reproject and estimate the ground truth 2D pose of the pitcher in camera coordinates from the annotated ground truth 3D pose data captured in the world coordinate frame. This introduces some additional constraints to estimate the exact mapping parameters between the 3D world coordinates and 2D camera coordinates which is required to train the 2D pose estimator. Consequently, certain assumptions and approximations were made to work around the limitations in this dataset to ensure its validity, and those components can be visualized in Figure 3.1. The proposed data processing module to circumvent the aforementioned challenges is detailed in the following section.

3.2 Dataset Processing

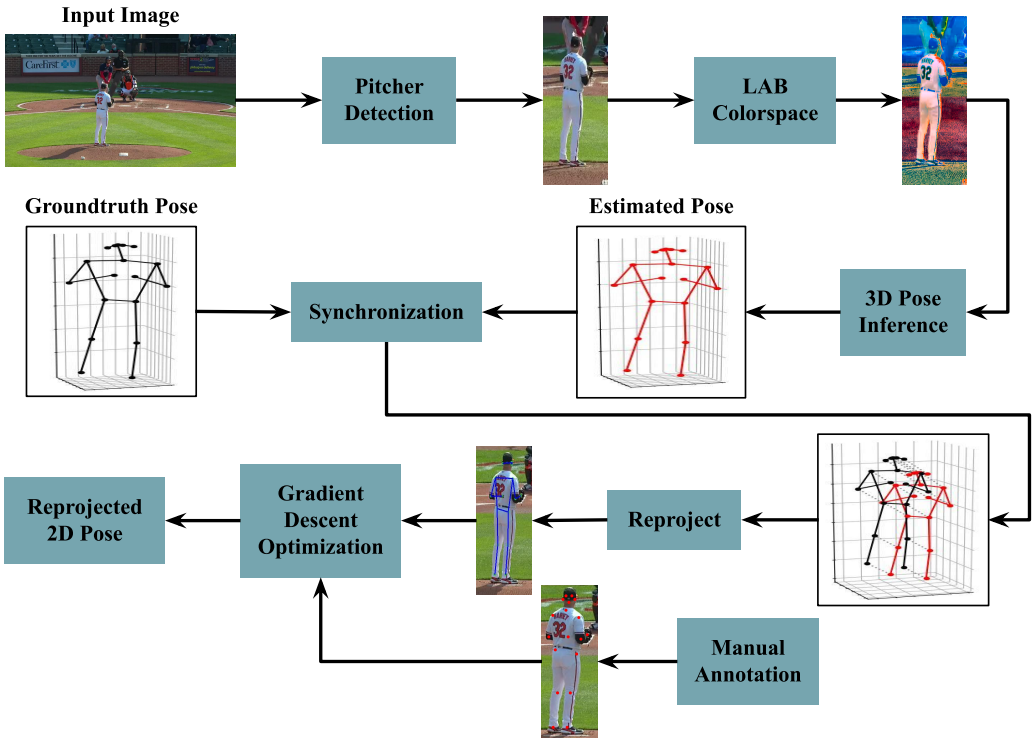


Figure 3.1: Overview of the data processing framework.

The data processing module addresses three primary objectives. The first objective tackles the task of improved player identification, specifically focusing on the pitcher. Accurate detection and isolation of the pitcher in each frame is crucial for accurate analysis. This is achieved through a player detection technique, which will be further detailed in Section 3.2.1. Additionally, image enhancement techniques, explained in Section 3.2.2, are employed to improve the pitcher's visibility within the frames. This combined approach ensures the data focuses on the player of interest and provides clear visual information for analysis.

Data consistency is the second objective addressed by the processing module. Missing annotations and 3D groundtruth pose misalignment issues with the broadcast frames hinders analysis. A time series technique, explained in Section 3.2.3, is employed to address these inconsistencies and ensure the data aligns seamlessly for further processing.

The final objective deals with the absence of camera parameters within the raw data. Camera parameters are critical for obtaining ground truth 2D pose information, which is essential for many machine learning tasks. A dedicated step, further detailed in Section 3.2.4, is implemented within the processing module to address this challenge.

By addressing these three objectives, the data processing module transforms the raw MLBPitchDB data into a high-quality and reliable dataset. This prepared dataset is then suitable for further analysis and machine learning tasks aimed at understanding and evaluating baseball pitching mechanics.

3.2.1 Pitcher Detection

Since our primary focus is to analyze pitching mechanics, accurately identifying and isolating the pitcher in each frame is crucial. This section details a three-step process implemented within the data processing module to achieve this objective.

The first step involves identifying keyframes within each player tracklet. Keyframes are frames that best represent the overall motion within a tracklet. Identifying these keyframes enables focusing our jersey number identification framework on the most informative moments within each pitching sequence. The second step utilizes a domain-guided masked autoencoder. It focuses on extracting informative spatial features from the identified keyframes. Finally, a temporal decoder takes the keyframe features extracted by the autoencoder and learns the temporal context of the entire tracklet.

Keyframe Identification

The Keyframe Identification (KfID) module helps detect and aggregate jersey number features based on their visibility, providing a *spatial-context* to the classification task. For a given player tracklet $\mathcal{T} = \{F_i : F_i \in \mathbb{R}^{H \times W \times 3}\}_{i=1}^{t}$ consisting of t frames, $KfID(\mathcal{T}) = \mathcal{T} \setminus \{F_{n_1}, F_{n_2}, ..., F_{n_k}\}$, where $F_{n_1}, F_{n_2}, ..., F_{n_k}$ are noisy frames with diminutive digit features. In a way, our KfID module works as a selective filter, eliminating those frames that biases our Spatio-Temporal Network towards inaccurate predictions due to inconsequential features.

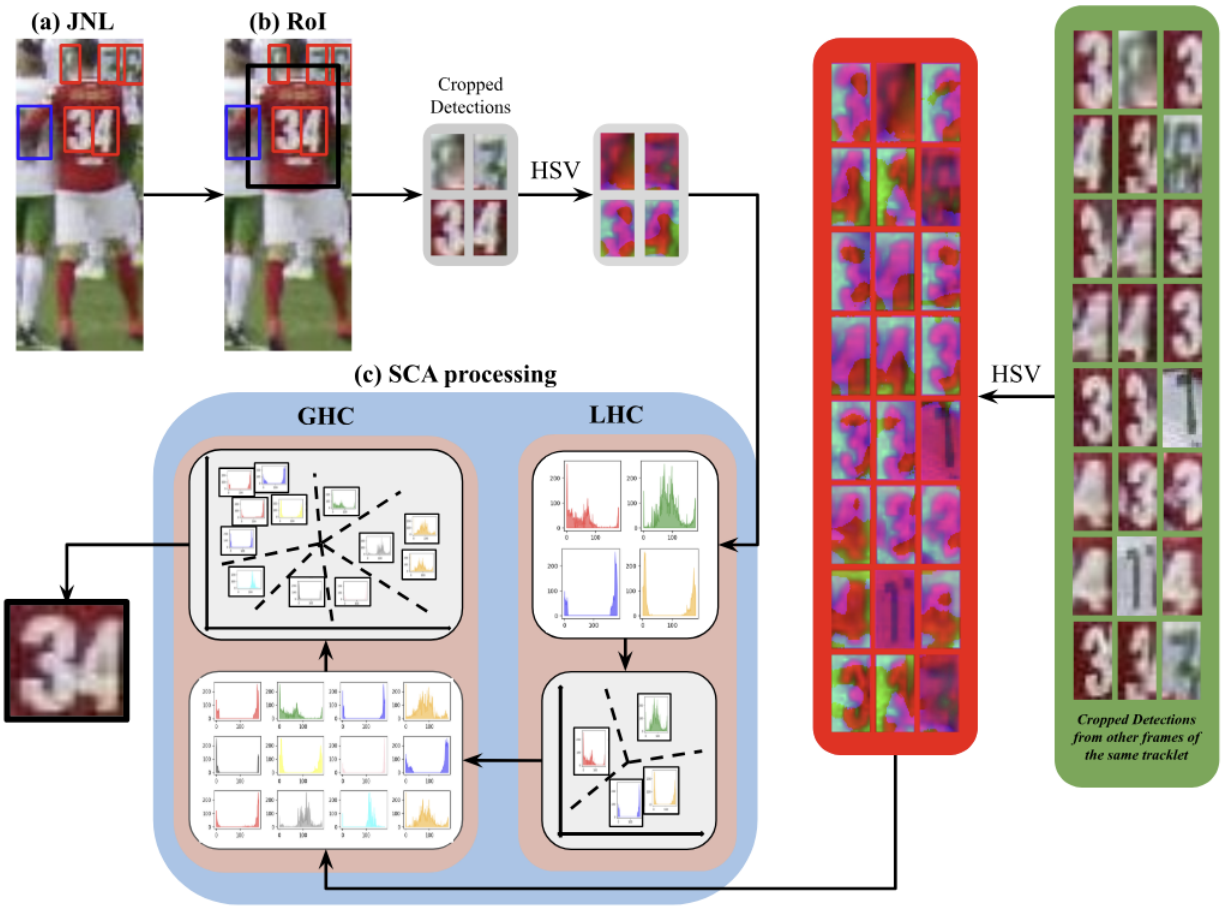


Figure 3.2: Overview of the proposed KfID module for jersey number-based keyframe extraction.

The pipeline for our KfID module is as follows: For each frame F_i of a given tracklet $\mathcal{T}$, the JNL module localizes all the digits in F_i, which are denoted as det_i, as shown in Equation (3.1). The detections within F_i are then locally filtered ($f_{local}^{(i)}$) using RoI and Local Histogram Correlation (LHC) modules as shown in Equation (3.2). The Global Histogram Correlation (GHC) module captures the spatial similarity of the detections across frames of $\mathcal{T}$ and further filters them to obtain the jersey numbers of our player of interest (f_{global}), as shown in Equation (3.3).

$$det(\mathcal{T}) = \{det_i\}_{i=1}^{t} = \{JNL(F_i)\}_{i=1}^{t}. \tag{3.1}$$

$$f_{local}(\mathcal{T}) = \{f_{local}^{(i)}\}_{i=1}^{t} = \{LHC(RoI(det_i))\}_{i=1}^{t} \tag{3.2}$$

$$f_{global}(\mathcal{T}) = GHC(f_{local}(\mathcal{T})) \tag{3.3}$$

The LHC module is used to obtain a holistic representation of the jersey number by comparing correlation scores within each frame of the tracklet. If two detections in the frame are in close proximity and demonstrate similar spatial layouts (as indicated by correlation scores), they are likely to correspond to two digits that form part of the same jersey number. Consequently, these detections are merged to create a holistic representation.

Following the LHC module, the GHC module addresses the issue of jersey numbers present on opposition player jerseys. We cluster the constructed histograms of all the filtered detections of a tracklet to find out spatially similar detections. The cluster with the most number of detections is chosen, and the detections in that cluster are passed on to the spatiotemporal network. By clustering the histograms based on distribution frequency within detections across all frames in a tracklet, we effectively model the spatial layout of player tracklets and distinguish the jersey number of the target player while filtering out unwanted detections within each tracklet.

Domain-guided Masked Autoencoders

The overview of the proposed masked autoencoders is shown in Fig. 3.3. Initially, the frames of a player tracklet are processed in the KfID module as detailed in Section 3.2.1, where keyframes crucial for identifying jersey numbers are extracted. Subsequently, these keyframes are passed to our d-MAE encoder, which captures features with rich semantic representations of each keyframe. These spatial features are then passed to the temporal transformer decoder, which extracts temporal cues and predicts the jersey number associated with the tracklet.

The application of Masked Autoencoders (MAE) becomes particularly significant for jersey number detection due to the dynamic nature of the game. Players are frequently in motion leading to challenges such as occlusion, low-visibility and motion blur. However, we recognize that the conventional masking policy (zero-out) used in MAE does not encompass the diverse conditions encountered in real-world scenarios. For instance, viewing an image through a window filled with water droplets provides a form of masking that is not similar to zeroing out image patches. Similarly, broadcast feeds of fast-paced sports like Soccer, Basketball, or Ice Hockey highlight the occurrence of such visual distortions and blurring effects that affect visual clarity.

Motivated by the need to recover missing or occluded spatial information to acknowledge these diverse scenarios, we build on the proficiency of MAE to reconstruct missing

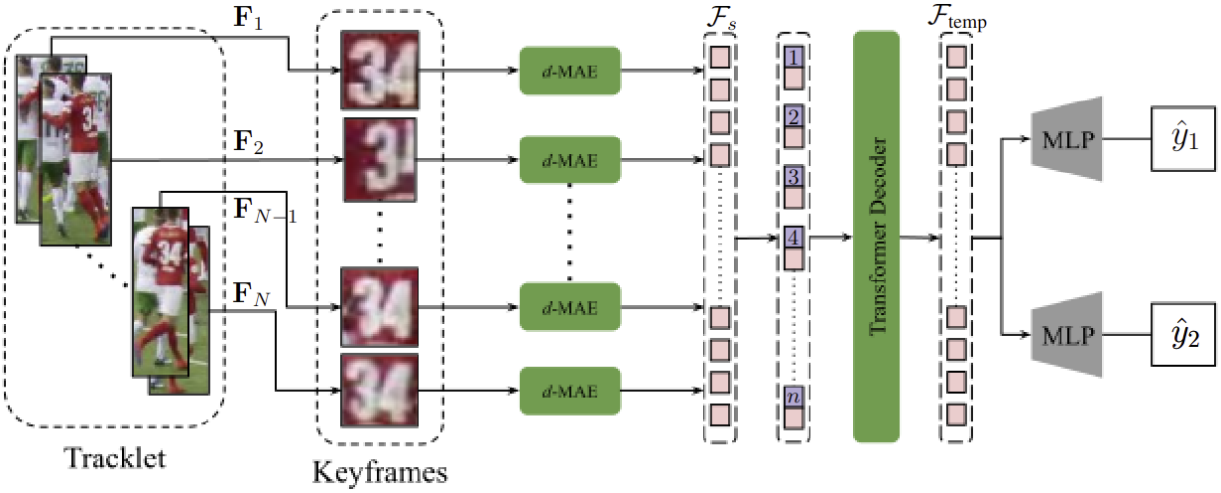


Figure 3.3: ***d*-MAE architecture**. Given a tracklet $\mathbb{T}$ consisting of N frames, we pass $\mathbb{T}$ through the KfID module to extract $n \leq N$ keyframes that contain the jersey number. Each keyframe is passed as an input to our d-MAE encoder to extract spatial features $\mathcal{F}_s$. These features are then fed to the temporal transformer decoder to extract temporal features $\mathcal{F}_{\text{temp}}$. Two classification heads are utilized to compute the predicted digits of the jersey number $\hat{y}_1$ and $\hat{y}_2$ respectively.

patches within the pixel space by introducing a domain-guided masking strategy. Particularly, during the pre-training stage, we incorporate motion blur to the patches instead of simply zeroing-out them, thereby infusing domain knowledge in the process. This approach facilitates reliable and accurate prediction of jersey numbers in dynamic sports scenarios. We incorporate an additional supervision to the pre-training objective, improving the feature extraction process.

Pre-training. The input image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ is split into K patches $\mathbb{I} = \{\mathbf{I}_k \in \mathbb{R}^{P^2 \times D}\}_{k=1}^{K}$ where P is the patch size, D is the embedding dimension and $K = HW/P^2$. A random subset of the patches $\mathbb{S} \subseteq \mathbb{I}$ are then masked by introducing motion blur artifacts to the pixels, resulting in the set

$$\mathbb{I}_{\text{masked}} = \{m(\mathbf{x}) : \mathbf{x} \in \mathbb{S}\},$$

where $m : \mathbb{R}^{P^2 \times C} \to \mathbb{R}^{P^2 \times C}$ is the mask applied to random patches. The unmasked patches $\mathbb{I}_{\text{unmasked}} = \mathbb{I} \setminus \mathbb{S}$ are then converted to unmasked tokens

$$\texttt{[unmask]} = \text{concat}(\mathbf{x} : \mathbf{x} \in \mathbb{I}_{\text{unmasked}})$$

and passed to the MAE encoder to extract latent spatial features

$$\mathcal{F}_s = f(\texttt{[unmask]}),$$

where $f : \mathbb{R}^{K \times P^2 \times D} \to \mathbb{R}^{K \times P^2 \times D}$ denotes the MAE encoder. The masked tokens

$$\texttt{[mask]} = \text{concat}(\mathbf{x} : \mathbf{x} \in \mathbb{I}_{\text{masked}})$$

are then used along with the latent features to generate the reconstructed image

$$\hat{\mathbf{I}} = g(\text{concat}(\texttt{[mask]}, \mathcal{F}_s)),$$

where $g : \mathbb{R}^{K \times P^2 \times D} \to \mathbb{R}^{H \times W \times C}$ denotes the MAE decoder.

To induce motion blur in the selected patches, we employ an oriented motion blur filter $\mathbf{K}'$ characterized by two parameters: the angle of rotation (ω) and the scale factor (s_f). The filter is centered at $(k_s/2, k_s/2)$ where k_s is the kernel size. Eq. (3.4) denotes the motion blur filter used to apply motion blur on image patches.

$$m(\mathbf{I_k})(x, y) = \frac{\mathbf{K}'}{\sum_{i,j} \mathbf{K}'_{ij}} * \mathbf{I}_k(x, y) \tag{3.4}$$

$$\mathbf{R} = \begin{bmatrix} s_f \cos(\omega) & -s_f \sin(\omega) & \frac{k_s}{2}(1-s_f \cos(\omega))+\frac{k_s}{2} s_f \sin(\omega) \\ s_f \sin(\omega) & s_f \cos(\omega) & \frac{k_s}{2}(1-s_f \cos(\omega))-\frac{k_s}{2} s_f \cos(\omega) \end{bmatrix} \tag{3.5}$$

where $*$ denotes the convolution operation and $m(.)$ is the masking strategy we employ at every pixel position (x, y) of an image patch $\mathbf{I}_k$. The rotation matrix $\mathbf{R}$ used to generate the oriented filter $\mathbf{K}'$ is shown in Eq. (3.5).

This tailored approach facilitates our d-MAE in capturing crucial cues necessary for the accurate reconstruction of the keyframes in the presence of motion blur.

Transformer Decoder

To capture the temporal cues within the tracklet, we extend our MAE module, by introducing a transformer decoder. Specifically, after the pre-training stage, we discard the decoder of d-MAE during finetuning, and pass the original unmasked keyframes directly to the d-MAE encoder. The extracted latent spatial features $\mathcal{F}_s$ are fed to the temporal transformer decoder to perform jersey number recognition. Leveraging the standard Vision Transformer (ViT) architecture for our decoder, we utilize self-attention to capture long-range dependencies between the spatial features of different frames within a player tracklet. By employing the self-attention mechanism on the spatial tokens, we facilitate the model's ability to understand the global context and intricate connections between keyframes of a tracklet. The resulting representation $\mathcal{F}_{\text{temp}}$, encapsulates rich cues on the jersey number, which are crucial for player identification.

3.2.2 Colorspace Conversion

The cropped frames with pitcher were then converted into the LAB colorspace which consists of two color channels and a luminosity channel. Then, contrast enhancement on the luminosity channel was done to increase the overall brightness level and amplify the adjacent pixel intensities. Then it was converted to BGR colorspace to improve the visual quality of the images while still preserving color information.

3.2.3 Data Synchronization

To synchronize between frames and the 3D ground truth pose data, Dynamic Time Warping (DTW) [101] was employed as a method of aligning the two sequences. By warping the

time axis and minimizing the distance or cost between the sequences, DTW finds the ideal alignment. Directly aligning an image with 3D keypoints is impractical; thus, a two-step approach was adopted. Firstly, an off-the-shelf 3D pose estimator was used to estimate the pose at each frame. Subsequently, DTW was used to find the best alignment between the GT 3D pose sequence and the estimated poses. Given the data and the constraints of the problem, a one-to-one relation was established as a hard constraint thereby enforcing unique correspondences between poses. Equation (3.6) in the following represents the cost function ($\mathcal{G}$) that was constructed as part of the alignment process and takes into account both the spatial and temporal components of the data.

$$\mathcal{G} = g_s \left(\frac{1}{\mathcal{J}} \sum_{i=1}^{\mathcal{J}} (kp_{gt}^{(i)} - kp_{pred}^{(i)})^2 \right) + g_t \left(1 - \frac{\sum_{i=1}^{\mathcal{J}} kp_{gt}^{(i)} \cdot kp_{pred}^{(i)}}{\sqrt{\sum_{i=1}^{\mathcal{J}} (kp_{gt}^{(i)})^2} \cdot \sqrt{\sum_{i=1}^{\mathcal{J}} (kp_{pred}^{(i)})^2}} \right) \tag{3.6}$$

Here, kp_{gt} and kp_{pred} are the ground truth keypoints and estimated keypoints respectively, and g_s and g_t correspond to the spatial and temporal weight gains, respectively. $\mathcal{G}$ is formulated to simultaneously account for spatial and temporal aspects of the pose data, providing a more accurate approach to align the data. The mean square error was used to capture the spatial discrepancy between keypoints and cosine similarity was utilized to measure the difference in angle between subsequent frames to calculate the temporal context of the pose data. The estimated spatial and temporal distances are then added and represented as bins of a histogram to then compare with other pose representations.

3.2.4 Camera Projection

To address the absence of camera parameters for mapping 3D world pose coordinates to 2D camera pose coordinates, we utilize an iterative optimization approach to find the optimized camera parameters. We begin by manually annotating the 2D pose in a reference frame and initializing a focal length. Through a process of gradient descent optimization, we iteratively refine the mapping by adjusting the focal length. This adjustment is performed to minimize the error between the projected 2D pose and the annotated 2D pose. Equation (3.7) outlines the optimization process used in this approach.

$$\hat{f} = f_i - \alpha \Delta L(f_i) \tag{3.7}$$

Here, $\hat{f}$ and f_i represent the updated and previous focal lengths respectively, and α and $\Delta L(f_i)$ represent the learning rate and the gradient of the loss function respectively.

During each iteration, the gradient of the loss function with respect to the focal length is computed. This gradient guides the adjustment of the focal length towards values that result in a more accurate 2D pose projection. By iteratively updating the focal length in the direction that reduces the loss, the mapping between 3D world coordinates and 2D camera coordinates is progressively refined. This iterative optimization scheme enhances the accuracy of the 3D-to-2D projection by effectively incorporating camera parameters.

3.3 Summary

To summarize, this chapter outlined the data processing module, a critical step in transforming the raw MLBPitchDB collection into a high-quality dataset suitable for analyzing baseball pitching mechanics. The module addressed three key challenges: player identification, data consistency, and the absence of camera parameters.

Firstly, a novel jersey number identification framework was implemented to isolate the pitcher tracklet within each video. This framework utilizes keyframe identification and masked autoencoders to extract relevant visual information from the most informative frames. Secondly, image enhancement techniques improved the visual clarity of the pitcher within each frame. Finally, data synchronization techniques and camera projection methods addressed inconsistencies and optimized the 2D pose information of the pitcher.

Through this multi-step processing pipeline, the data processing module successfully transformed the raw data into a reliable and informative dataset. This prepared dataset lays the groundwork for further analysis and evaluation of the baseball pitching mechanics.

Chapter 4

Distribution- and Depth-Aware 3D Human Modeling

4.1 Overview

Despite the notable progress in monocular HMR, they struggle with two key challenges-appearance domain gap and depth ambiguity. Controlled environments, often used for training, offer a setting where data collection and annotation are manageable and precise. However, the challenge arises when the trained model is applied to in-the-wild data, where real-world variability, such as lighting conditions, backgrounds, and poses, differs significantly from controlled settings. Second, depth-ambiguity issues plague single-view images. In response to the latter challenge, researchers, as exemplified in [53] and [49], have proposed solutions that leverage temporal information extracted from video inputs to enhance the understanding of human motion. However, these temporal approaches have entailed significant computational overhead.

Obtaining ground truth mesh labels for human mesh reconstruction is a tedious task, mainly due to challenges like complexities of dynamic human motion, scene dynamics, resource constraints, and privacy concerns. In response to the inherent difficulty in obtaining accurate ground truth labels, existing works such as [70, 39, 28, 55] resort to using pseudo ground truth to train models. Consequently, the modeling of human forms is inherently biased due to the presence of noisy labels. Moreover, the generalization of HMR for Out-Of-Distribution (OOD) poses, as discussed earlier, presents an immensely challenging problem. Prior works [59, 130] model the output as a distribution of plausible 3D pose using normalizing flows and use information such as 2D keypoints or part segments

as priors to provide deterministic predictions for downstream tasks. However, since these models use normalizing flows to estimate the underlying output distribution, they do not predict out-of-distribution data as shown in [52] and do not solve the model bias to actual data, especially in scenarios with noisy labels and uncertainties.

Against this backdrop, in this chapter, we introduce a novel approach to tackle these issues through a depth and distribution-aware framework designed for the recovery of human mesh from monocular images. Notably, we explicitly integrate scene-depth information from monocular cameras obtained from prior depth models (termed as *pseudo-depth*) into a transformer encoder via the cross-attention mechanism. Moreover, we employ a residual log-likelihood approach to learn deviations in the underlying distribution, facilitating a refinement module in the training process. To further refine the mesh shape and feature relationships, we introduce a dedicated silhouette decoder and a masked modeling module. To the best of our knowledge, D2A-HMR is the only framework to explicitly incorporate depth priors and systematically learn the mesh distribution disparity between the underlying prediction and ground truth distributions.

4.2 Preliminary

4.2.1 Normalizing flow

Normalizing flow is a tool to efficiently transform a simple distribution into a complex one through a series of invertible transformations [52, 59]. It applies to probability density estimation, which can be used to estimate the likelihood. Previous work including [130] and [12] uses normalizing flows to learn a prior on distributions of plausible human poses. ProHMR [59] focuses on modeling the output of the human mesh as a distribution over all the different possible meshes. However, it utilizes normalizing flows to directly predict the exact underlying distribution which is demonstrated to perform poorly for OOD data [52]. RLE [64] uses normalizing flow to minimize the difference between the distributions of the ground-truth and predicted 2D poses rather than using the output distribution to sample one particular pose, thereby boosting the performance of regression-based pose estimation techniques.

Inspired by the literature on residual log-likelihood in 2D human pose estimation [64] and the shortcomings of existing HMR approaches, our approach focuses on mitigating distribution discrepancies of the output and ground truth meshes by leveraging normalizing flow techniques. This alleviates the problem of poor performance on OOD data as we

use normalizing flows in the refinement module to minimize the difference between the output mesh distribution and ground truth mesh distribution instead of predicting output poses/meshes using the captured output distribution.

4.2.2 Attention for Human Mesh Recovery

Attention mechanisms have been shown to be effective for HMR by enabling models to focus on the most relevant parts of the input data. METRO [70] uses self-attention to reduce ambiguity by establishing non-local feature exchange between visible and invisible parts with progressive dimensionality reduction. SAHMR [105] uses cross-attention between image and scene contact information to improve the posture of the regressed mesh. The recently proposed JOTR [63] uses self-attention to study the 2D and 3D feature dependencies to tackle issues with occlusion. PSVT [97] uses a spatio-temporal attention mechanism to capture relations between tokens and pose/shape queries in both temporal and spatial dimensions. Similarly, OSX [69] uses a component-aware encoder to capture the correlation between different parts of the human body to predict the whole-body human mesh.

We propose a parallel network composed of two self-attention modules to learn global dependencies within the image and pseudo-depth features, respectively, and a cross-attention module to learn inter-modal dependencies between the image and pseudo-depth features. This allows the network to learn a more comprehensive representation for accurate 3D mesh recovery.

4.3 Methodology

The overview of the proposed D2A-HMR framework is presented in Figure 4.1. In this section, we delve into the architecture and training objective of D2A-HMR. The feature encoding process begins with the extraction of features from the image and pseudo-depth map using a CNN backbone, followed by hybrid position encoding. These encoded features are then input to the transformer encoder, which engages in cross-attention with the pseudo-depth cues and the input image. Following this, the refinement module comes into play, incorporating the distribution matching, silhouette decoder, and masked modeling components to regularize the model during the training process.

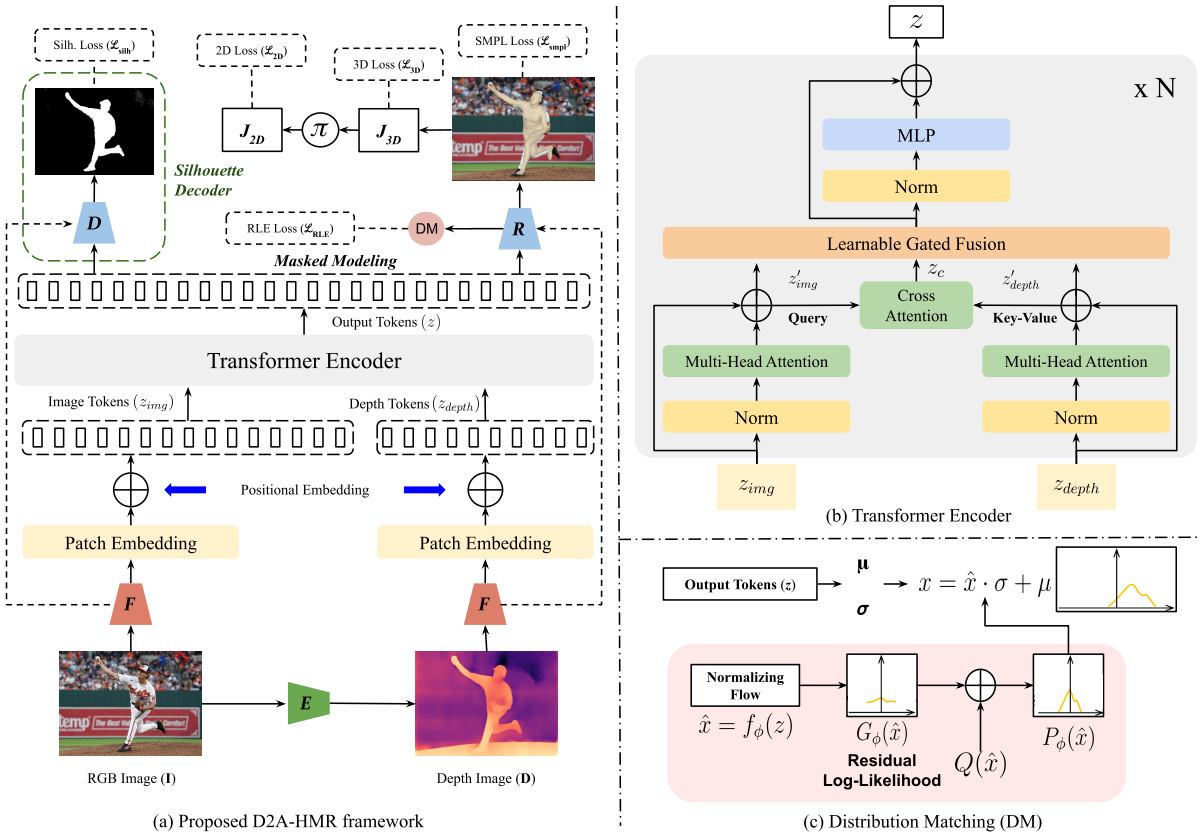


Figure 4.1: D2A-HMR model architecture

4.3.1 Problem Statement

Given an image (**I**), we first incorporate a transformer backbone (***E***) to estimate the depth map (**D**) and a CNN backbone (***F***) to extract the features from the images. Positional embedding is applied to both image and pseudo-depth features, utilizing a hybrid approach for image tokens (z_{img}) and a learnable position embedding for pseudo-depth tokens (z_{depth}). Self-attention is performed on z_{img} and z_{depth}, resulting in z'_{img} and z'_{depth}, respectively. Subsequently, cross-attention is applied between z'_{img} and z'_{depth} to produce z_c. The learnable fusion gates combine z'_{img}, z'_{depth}, and z_c, followed by layer normalization and an Multilayer Perceptron (MLP). The resulting gated tokens (z) are input into two distinct refinement modules: a decoder (***D***) for silhouette estimation and a regressor head, ***R*** which incorporates normalizing flow (**DM**) for distribution-aware joint vertex estimation.

4.3.2 Architecture

Feature Encoding. The initial step involves passing the input image and depth map through a CNN backbone to extract pertinent features. Subsequently, to explicitly model the structure of the features, position embedding is applied to these extracted features. Specifically, we implement a hybrid positional encoding (P_e) illustrated in Equation (4.1), for the image tokens. This hybrid approach capitalizes on the strengths of both learnable position embeddings (P_l) and sinusoidal position embeddings (P_s). P_l adapts to task-specific positional patterns, proving highly effective in capturing intricate spatial relationships. Meanwhile, P_s contributes to the globally consistent positional understanding, capturing more information about the position. This combination optimally balances adaptability and global context, yielding fine-grained spatial patterns and general positional relationships.

$$P_e = \omega_1 P_l + \omega_2 P_s \tag{4.1}$$

where ω_1 and ω_2 are learnable parameters controlling the position embedding contribution of both types.

Transformer Encoder. The utilization of the transformer encoder in D2A-HMR is driven by the overarching goal of effectively learning pseudo-depth cues from the input data. Using self-attention mechanisms on the encoded features derived from both modalities (image and pseudo-depth map), namely z_{img} and z_{depth}, the transformer encoder facilitates understanding of spatial relationships within each domain. Furthermore, we propose to use a cross-attention mechanism to establish intricate connections between the image and pseudo-depth information. The resulting fused representation, denoted as z, encapsulates rich depth cues, crucial for the subsequent regression of human vertices.

The embedded features, denoted as z_{img} and z_{depth}, serve as input tokens to the transformer encoder, embodying our pursuit of learning pseudo-depth cues. Using self-attention mechanisms, the encoder refines z_{img} and z_{depth} by capturing spatial relationships within each modality, producing updated features z'_{img} and z'_{depth}, respectively. Subsequently, the introduction of a cross-attention mechanism facilitates connections between image and pseudo-depth features. The resulting cross-attended tokens denoted as z_c, are then fused with z'_{img} and z'_{depth} from their respective attention heads, yielding a final fused representation denoted as z, as illustrated in Equation (4.2). To facilitate this fusion, learnable fusion gates are employed, similar to the position encoding methodology. These gates

adaptively emphasize the importance of each source, enhancing the model's capacity to capture meaningful relationships between the image and pseudo-depth features.

$$z = \omega_3 z'_{img} + \omega_4 z'_{depth} + (1 - \omega_3 - \omega_4) z_c \tag{4.2}$$

Here, in Equation 4.2, ω_3 and ω_4 are the learnable parameters. Once the fusion is done, z is normalized and fed as input to an MLP to get the output tokens. This holistic approach enables our model to effectively capture intricate patterns and dependencies within the input image and the 3D information of the scene. A visual illustration of the transformer encoder is shown in Figure 4.1.

4.3.3 Refinement Module

The refinement module in the D2A-HMR framework encompasses three key components, each designed to enhance the model's capabilities in capturing different aspects of human pose and shape. First, the distribution matching component aids in refining the model's representation by aligning the output mesh distribution to the ground truth mesh distribution. This adaptation enables the model to capture and adapt to inherent variations in the distribution of training data, promoting a more generalized performance that extends beyond the specific characteristics of the training data. The second component, the silhouette decoder, focuses on optimizing the model's capacity to align the shape with the input image by adeptly capturing the outlines of the human subject. This component contributes significantly to the model's ability to refine and improve its representation based on the visual cues present in the input data. Lastly, the masked modeling component serves to empower the model by learning from available information, thereby enhancing its ability to capture long-range relationships among features in the image. This integration ensures that the model can leverage relationships across the entire input, contributing to a more comprehensive understanding of the underlying human pose and shape.

Distribution Matching. To align the model closely to the actual underlying data distribution, we incorporate the normalizing flow mechanism proposed by [32] within the D2A-HMR framework. Our goal is to refine the model by learning the disparity between the predicted and groundtruth distributions. The output tokens z from the transformer encoder are passed via a regressor ($\boldsymbol{R}$) to find the standard deviation σ and the mean μ which control the scale and position of the distribution respectively. It is used to transform the initially assumed Gaussian distribution. The distribution modeled by the flow ($P_\phi(\hat{x})$) is deconstructed into three essential terms, as expressed in the equation:

$$\log P_{\phi}(\hat{x}) = \log Q(\hat{x}) + \log \frac{P(\hat{x})}{c \cdot Q(\hat{x})} + \log c \tag{4.3}$$

The first term, $\log Q(\hat{x})$, quantifies the logarithmic probability of the data under the simple distribution. The second term, $\log \frac{P(\hat{x})}{c \cdot Q(\hat{x})}$, represents the residual log-likelihood, serving as the distinction between the log-probability of the data under the optimal underlying distribution and the log-probability under the tractable initial density function. The third term, $\log c$, functions as a normalization constant.

Silhouette Decoder. To optimize shape alignment, we used a specialized decoder to reconstruct silhouettes. Leveraging features from the transformer encoder, this decoder employs a sequence of deconvolution layers with ReLU activation and dropout, culminating in a fully connected layer. This intricate reconstruction process significantly augments the model's capability to generate high-quality silhouette representations. To acquire the pseudo-ground truth silhouette of human subjects, we utilize an existing segmentor [71].

Masked Modeling. Prior works including [31], [70], and [94] have demonstrated the efficacy of masked modeling in elucidating diverse relationships within training datasets, spanning textual, vertex, and image domains respectively. In alignment with these established works, we adopt random masking of the embedded features to recover the vertex of the human body. By deliberately obscuring a percentage of embedded features during training, our model is forced to rely solely on the unmasked features extracted from the image. This enables a comprehensive understanding of both short and long-range relationships among the features, contributing to the overall performance of D2A-HMR framework.

4.3.4 Loss Functions

In this sub-section, we present the comprehensive training objectives employed to recover the human mesh in our model. These objectives consist of a weighted combination of various loss components, each serving a specific role in refining the model's output.

The loss functions $\mathcal{L}_v$ and $\mathcal{L}_j$ are computed using the L_1 loss metric, aiming to minimize the disparities between the model's output vertices and the 3D human pose coordinates with the ground truth vertice and pose representation. Simultaneously, $\hat{\mathcal{L}}_j$ leverages the

same loss metric to optimize the 3D pose by regression of the output mesh vertices following [70], seeking alignment with the ground truth pose coordinates. To enhance the alignment between image and mesh representations, camera parameters are employed to reproject and infer the 2D human pose coordinates represented with $\mathcal{L}'_j$. This reprojected output is refined by applying loss optimization using L_1.

As mentioned in Section 4.3.3, a distribution matching regularizer is used to penalize the model for predicting outputs that are unlikely under the underlying ground truth distribution. Equation (4.4) shows the distribution regularizer ($\mathcal{L}_{RLE}$) used in the D2A-HMR architecture.

$$\mathcal{L}_{RLE} = -\log Q(\bar{\mu}_g) - \log G_\phi(\bar{\mu}_g) - \log c + log\ \sigma \tag{4.4}$$

Here, in Equation (4.4), $G_\phi(\bar{\mu_g})$ is the learned residual distribution of the predicted value $\bar{\mu_g}$ where $\bar{\mu_g} = (\mu_g - \mu)/\sigma$. Here, μ_g is the ground truth deviation and ϕ is the flow model parameter. We also incorporate silhouette loss, denoted as $\mathcal{L}_{silh}$, which regularizes the model by controlling the shape of the reconstructed mesh. The overall objective function is shown in Equation (4.5), which represents a combination of these individual losses.

$$\mathcal{L} = \lambda_d \mathcal{L}_{RLE} + \lambda_v \mathcal{L}_v + \lambda_{3D}(\mathcal{L}_j + \hat{\mathcal{L}}_j) + \lambda_{2D}\mathcal{L}'_j + \lambda_s \mathcal{L}_{silh} \tag{4.5}$$

where λ_d, λ_v, λ_{3D}, λ_{2D} and λ_s denote the weights attributed to the training objectives concerning the distribution, vertices, 3D pose coordinates, 2D pose coordinates, and silhouettes, respectively.

4.4 Experimentation

4.4.1 Implementation Details.

Training Details. Training was carried out on an infrastructure comprising three NVIDIA A6000 GPUs. The network was trained for 500 epochs, with a batch size of 48, and 24 parallel workers. Adam Optimizer, configured with a learning rate of 10^{-4} and beta values of 0.9 and 0.99, was used for optimization. The network was designed to output a coarse mesh representation containing 431 vertices. This output was subsequently upsampled [58] to the original mesh's 6890 vertices, utilizing learnable MLP layers, resulting in the model's ability to capture fine-grained spatial details.

Method	3DPW			Human3.6M	
	mPVE ↓	mPJPE ↓	PA-mPJPE ↓	mPJPE ↓	PA-mPJPE ↓
HMMR [49] (†)	139.3	116.5	72.6	-	58.1
SPIN [55]	116.4	96.9	59.2	62.5	41.1
TCMR [27] (†)	111.5	95.0	55.8	62.3	41.1
PyMAF [131]	110.1	92.8	58.9	57.7	40.5
ProHMR [59]	109.6	95.1	59.5	-	41.2
ROMP [112]	105.6	89.3	53.5	-	-
VIBE [53] (†)	99.1	93.5	56.5	65.6	41.4
I2LMeshNet [87]	-	93.2	57.7	55.7	41.1
METRO [70]	**88.2**	77.1	47.9	**54.0**	**36.7**
Pose2Mesh [28]	-	89.2	58.9	64.9	47.0
PARE [54]	88.6	**74.5**	**46.5**	-	-
D2A-HMR (Ours)	93.7	88.5	53.4	56.8	40.2

Table 4.1: Comparison to state-of-the-art 3D pose reconstruction approaches on 3DPW and Human3.6M datasets. (†) are temporal methods. **Bold**: best; Underline: second best; Double Underline: third best.

4.4.2 Main Results

We assess the performance of the proposed D2A-HMR framework by comparing it with established state-of-the-art techniques for HMR. The results, presented in Table 4.1, highlight the competitive performance of our method across various metrics on the Human3.6M and 3DPW datasets.

To further emphasize the efficacy of our proposed approach, we conducted a qualitative comparison against several state-of-the-art techniques, as depicted in Figure 4.2. These techniques include SPIN [55], PARE [54], METRO [70], ROMP [112], and PyMAF [131]. The comparative results clearly demonstrate that the meshes generated by the D2A-HMR framework exhibit superior alignment with the input image. Our method's adept understanding of pseudo-depth cues, distribution, and the person's silhouette contributes significantly to improved alignment, particularly in handling challenging input scenarios characterized by depth ambiguities and extreme poses. This underscores the robustness of our proposed method for handling complex input conditions.

Method	Acc. ↑	mPJPE ↓
METRO [70]	81.5	37.8
SPIN [55]	84.7	32.1
PARE [54]	84.0	33.7
D2A-HMR (Ours)	**87.9**	**30.6**

Table 4.2: Comparison of D2A-HMR on a baseball dataset [16]

Table 4.2 presents a comprehensive comparison between our proposed method and the established state-of-the-art HMR techniques, utilizing the baseball dataset [16]. Notably, D2A-HMR demonstrates superior performance in terms of accuracy and mPJPE on this dataset, which is characterized by high player motion blur and instances of self-occlusion. Qualitative results that show the effectiveness of our approach in handling the complexities posed by this specific dataset can be visualized in Table 4.3.

The provided results in Figure 4.4 showcase the qualitative performance of the D2A-HMR approach across Common Objects in COntext (COCO) and various sports datasets. These visualizations underscore the effectiveness and robustness of our technique in achieving accurate alignment with input images, particularly in challenging in-the-wild scenarios.

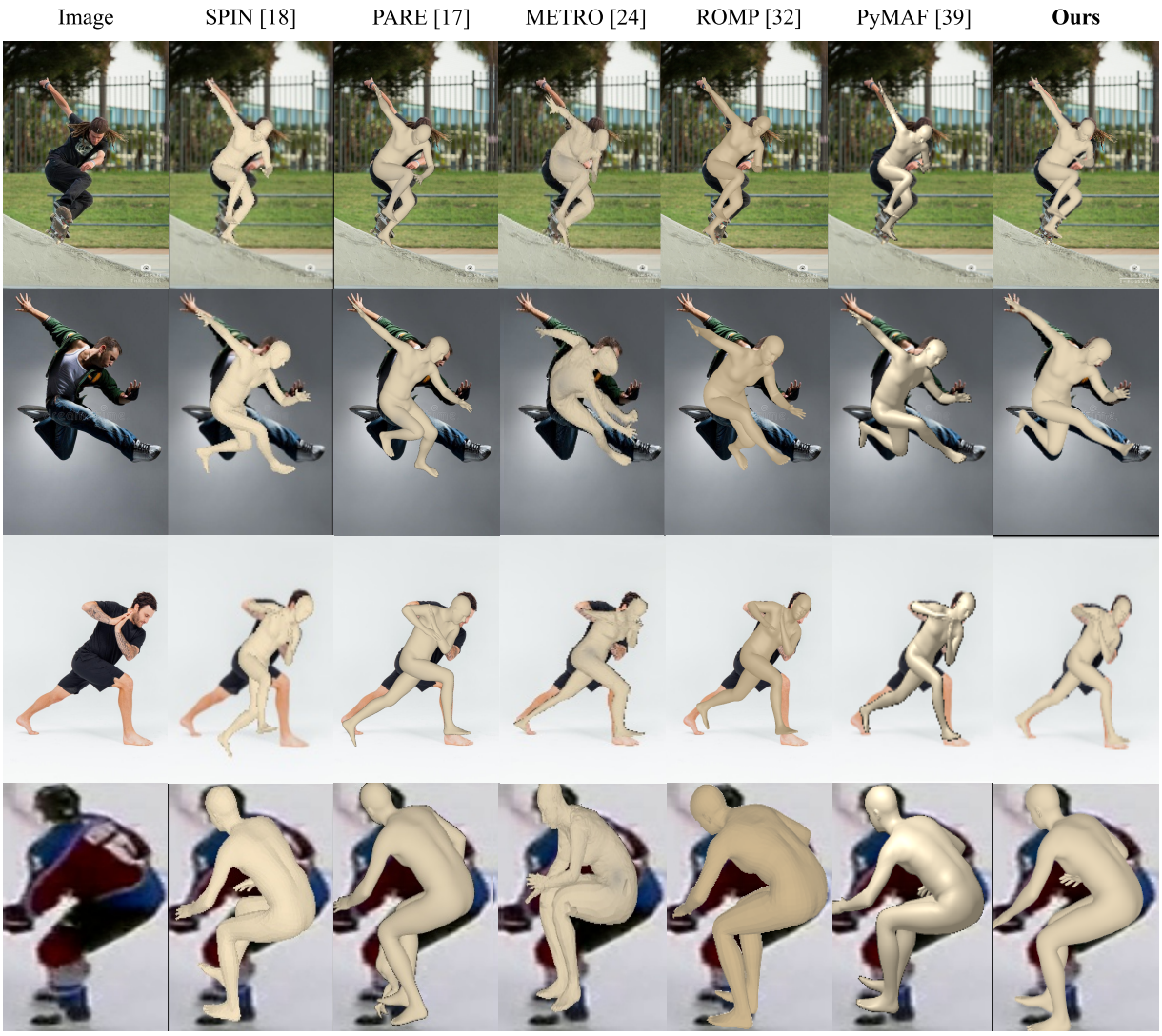


Figure 4.2: **Qualitative results.** Qualitative comparison of D2A-HMR with SPIN [55], PARE [54], METRO [70], ROMP [112] and PyMAF [131] on in-the-wild data from different sports dataset [16, 33, 47] and unusual poses from the internet.

Figure 4.3: **Qualitative results.** Inferred 3D mesh of D2A-HMR against some state-of-the-art HMR techniques on the baseball dataset [16].

Figure 4.4: **Qualitative results.** Qualitative comparison of D2A-HMR on COCO and sports datasets with unusual poses.

4.4.3 Ablation Studies

To verify the individual impact of each module on the proposed D2A-HMR model, comprehensive studies were conducted, as detailed in this sub-section. For consistency across all studies, the 3DPW dataset was utilized as the common benchmark.

Integration of multi-modal data. Experimentation to assess the impact of depth and distribution matching components within the D2A-HMR are detailed in Table 4.3.

Depth	Dist.	mPJPE ↓	PA-mPJPE ↓
✓		92.7	61.8
	✓	90.0	56.9
✓	✓	**88.5**	**53.4**

Table 4.3: Ablation study on pseudo-depth and distribution modeling for D2A-HMR evaluated on 3DPW dataset

Incorporation of both the pseudo-depth and distribution modeling modules in the D2A-HMR framework is observed to lead to a substantial improvement in the overall performance of mesh recovery. This observation serves as confirmation that the underlying motivation behind the proposed framework is valid and aids in enhancing the model's capabilities.

Depth on mPJPE(z). Experimentation on exclusively capturing the depth component of the regressed 3D joints in order to demonstrate its impact on the human pose was conducted in Table 4.4.

	mPJPE(z) ↓	PA-mPJPE(z) ↓
w/o depth modeling	69.1	58.3
w/ depth modeling	**65.4**	**53.6**

Table 4.4: Ablation study on the impact of depth modeling for D2A-HMR evaluated on 3DPW dataset

A notable enhancement in the z-axis of the reconstructed mesh is evident, as highlighted in Table 4.4. We computed mPJPE along the z-axis denoted as mPJPE(z), disregarding

the components x and y of the reconstructed mesh. This experimentation validates that the incorporation of scene-depth information contributes to an improvement in HMR.

Decoder and Masking Modules. Table 4.5 illustrates the impact of the silhouette decoder and masked modeling employed within the D2A-HMR framework.

Decoder	Mask	mPJPE ↓	PA-mPJPE ↓
✓		98.5	67.2
	✓	91.7	58.4
✓	✓	**88.5**	**53.4**

Table 4.5: Ablation study on the silhouette decoder and masked modeling for D2A-HMR evaluated on 3DPW dataset

The observations drawn from Table 4.5 highlight the beneficial impact of incorporating both the silhouette decoder and masked modeling modules in enhancing the model's ability to disentangle the appearance and part-relationship of the person. These modules are exclusively utilized during the training process of the D2A-HMR framework, contributing to its improved performance.

Backbones. A comprehensive analysis of D2A-HMR's performance by investigating its behavior with various backbone architectures was conducted. To establish a strong baseline, we first trained two ResNet variants for 1000 epochs on the ImageNet dataset [30] for an image classification task. We also explored HRNet variants trained for 1000 epochs using the COCO dataset [73] for the image classification task.

Backbone	mPJPE ↓	PA-mPJPE ↓
ResNet50	95.1	63.9
ResNet101	93.5	60.8
HRNet-w40	90.2	55.1
HRNet-w64	**88.5**	**53.4**

Table 4.6: Different input representations as the backbone for D2A-HMR evaluated on 3DPW dataset

We notice that HRNet-w64 gives the most positive impact on feature extraction from both the image and depth maps compared to the ResNet backbones. This can be attributed to HRNet-w64's effectiveness in capturing both local and global contexts through its multi-resolution fusion representations, thereby enhancing the model's ability to extract rich and informative features.

4.5 Summary

In this chapter, we introduced the Distribution and Depth-Aware Human Mesh Recovery (D2A-HMR) framework as an innovative solution to the persistent challenge of depth ambiguities and distribution disparities in monocular human mesh recovery. By explicitly incorporating scene-depth information, we have substantially reduced the inherent ambiguity, resulting in a more precise and accurate alignment of human meshes. The utilization of normalizing flows to model the output distribution has been instrumental in regularizing the model to minimize the underlying distribution disparities, enhancing its resilience against noisy labels, and mitigating biases in human-form modeling.

Our extensive experimentation on diverse datasets has demonstrated the competitive performance of the D2A-HMR method when compared to state-of-the-art HMR techniques. Furthermore, it has been noticed that our network outperforms existing work on sports datasets with OOD data. This proposed framework not only addresses depth ambiguities and mitigates noise, but also leverages the inherent 3D information present in images, providing a robust and unambiguous solution for human mesh recovery.

Chapter 5

Mitigating Motion Blur for Player Pose Modeling

5.1 Overview

A key challenge in pose estimation of humans with agile actions from broadcast videos is the quality of the input image, as factors such as motion blur and self-occlusion can degrade the performance of the reconstruction. In Figure 5.1, we illustrate an example of the challenges posed by a substantial motion blur effect during the pitching action, coupled with self-occlusion from the homeplate view. These issues underscore the complexity of the task and emphasize the need for robust and sophisticated methods to address such inherent limitations in the data. While some prior works have addressed motion blur caused by the camera [26, 109, 1], the problem of human-articulated motion blur remains largely unexplored. The impact of such motion blur on human pose estimation is significant. Significant advances have been made in tackling this issue [133, 81, 102], however, challenges persist when dealing with dynamic backgrounds or fast-moving human motions.

Figure 5.1: Sequence [86] captured at 30 fps from behind the homeplate view.

Thus, in this chapter, we present a unique approach for accurate pose estimation of pitchers in baseball games, considering the challenges addressed previously. Unlike existing methods that rely on complex pipelines, we propose a strategy centered on smart augmentation effects. By augmenting the training data with selective motion blur effects, we enhance the network's ability to learn and adapt to these effects. The inclusion of in-the-wild data from the Internet significantly bolstered the network's generalization capabilities to different camera positions and lighting conditions. This chapter highlights the effectiveness of a focused augmentation strategy and challenges the conventional notion of complex pipelines to handle motion blur in the sport of baseball.

5.2 Preliminary

5.2.1 Vision for Sports Analytics

In recent years, significant advancements have been made in sport-related pose estimation and body modeling techniques, which have greatly contributed to accurate performance analysis and understanding of human movement in sports. These techniques address common challenges faced in vision-based sports analytics, such as blur and occlusion. One approach, proposed by [102], presents a unified framework that combines deblurring and holistic 3D human body reconstruction. When the human reconstruction module is integrated, the deblurring module benefits from the human reconstruction loss, resulting in improved performance. Another approach, introduced by [81], focuses on a newly generated blurry human dataset and localized adversarial modules. Although these techniques have demonstrated significant improvements, they still encounter limitations, particularly in scenarios with dynamic backgrounds and significant motion differences between frames. Further research is needed to overcome these challenges and advance the effectiveness of vision-based sports analytics.

5.2.2 Mitigating Motion Blur

Most research on motion blur tackles camera-induced blur [26, 109, 1]. This is because camera blur affects the entire image uniformly, making it easier to model and potentially remove. Human motion blur, however, is far more challenging. Unlike camera blur, it varies across the image depending on the person's movement and scene complexity. This scene-dependence makes it difficult to isolate and address.

While limited research tackles human motion blur [100, 81, 102], some approaches show promise. FrankMocap [100] introduces random motion blur during training for 3D hand pose estimation. This helps their model achieve robust hand reconstruction, especially from real-world videos. Lumentut et al. [81] propose an end-to-end framework that combines deblurring with 3D pose estimation. D2R [102] employs a similar strategy, utilizing a specialized deblurring module within a holistic HMR technique. Both frameworks leverage the deblurred image to estimate accurate SMPL parameters.

5.3 Methodology

The proposed approach comprises several key steps aimed at enhancing the motion blur effect and estimating the 3D body model of the pitcher. Each pitch sequence is represented by a set $\hat{\mathcal{P}} = \{\mathcal{F}_t : \mathcal{F}_t \in \mathbb{R}^{H \times W \times 3}\}_{t=1}^{t_n}$. To augment the motion blur effect, the approach utilizes a motion blur learning module, where pairs of subsequent frames are taken for motion flow analysis. Each frame is divided into k patches of equal size from which $\mathcal{N}$ patches are selected to induce motion blur. The motion flow vector $\mathcal{M}_k^{(t)}$ of each patch is then estimated, which is denoted as $\mathcal{M}_k^t = \sum_{i,j=0,0}^{i,j=H,W} v_{ij}$ where v_{ij} is the flow vector for each patch at pixel position (i, j). Then, $\mathcal{N}$ patches with most $\mathcal{M}_k^t$ value is selected as the target regions to introduce motion blur effect.

Next, the 2D pose of the pitcher in each frame $\mathcal{F}_t$ is estimated, where the input is a frame containing the pitcher, and the output is a pose representation denoted as $\mathcal{P}_{2D}^{(t)} \in \mathbb{R}^{\mathcal{J} \times 2}$, where $\mathcal{J}$ represents the total joints of the pitcher. Following the 2D pose estimation, the 3D pose of the pitcher is estimated by utilizing s consecutive sets of 2D pose data as the receptive field, where the input is denoted as $\mathcal{P}_{2D} \in \mathbb{R}^{s \times \mathcal{J} \times 2}$. The output of the 3D pose estimation is a pose representation that is denoted as $\mathcal{P}_{3D} \in \mathbb{R}^{1 \times \mathcal{J} \times 3}$.

The 2D and 3D poses are then concatenated to form the input for the 3D body model, denoted as $\mathcal{P}_{\text{concat}}^{(t)} \in \mathbb{R}^{1 \times \mathcal{J} \times 5}$, which is the concatenation of its corresponding $\mathcal{P}_{2D}$ and $\mathcal{P}_{3D}$. The output of the 3D body model is represented as $\mathcal{H}_{3D} \in \mathbb{R}^{\mathcal{V} \times 3}$, where $\mathcal{V}$ represents the human vertices of the mesh.

5.3.1 Motion Blur Learning Module

The motion blur learning module aims to address the motion blur challenges of the dataset by augmenting the dataset with extra synthetic data mimicking it in a realistic manner. It essentially provides the network with different opportunities to see and learn from different instances of motion blur, by increasing the frequency and consistency of the effects, thereby increasing the robustness of the challenge.

To achieve realistic synthetic effects, our approach comprises a series of deliberate steps. Initially, we estimate the motion flow vectors between consecutive pairs of images. Subsequently, we integrate a two-step process to discern the specific regions (patches) where motion blur should be induced. This process involves a selective identification of patches that exhibit significant motion, ensuring a targeted approach to motion blur application.

The processed consecutive images from the dataset processing module (Section 3.2) are passed through a motion flow estimation algorithm proposed by [46]. It uses a transformer

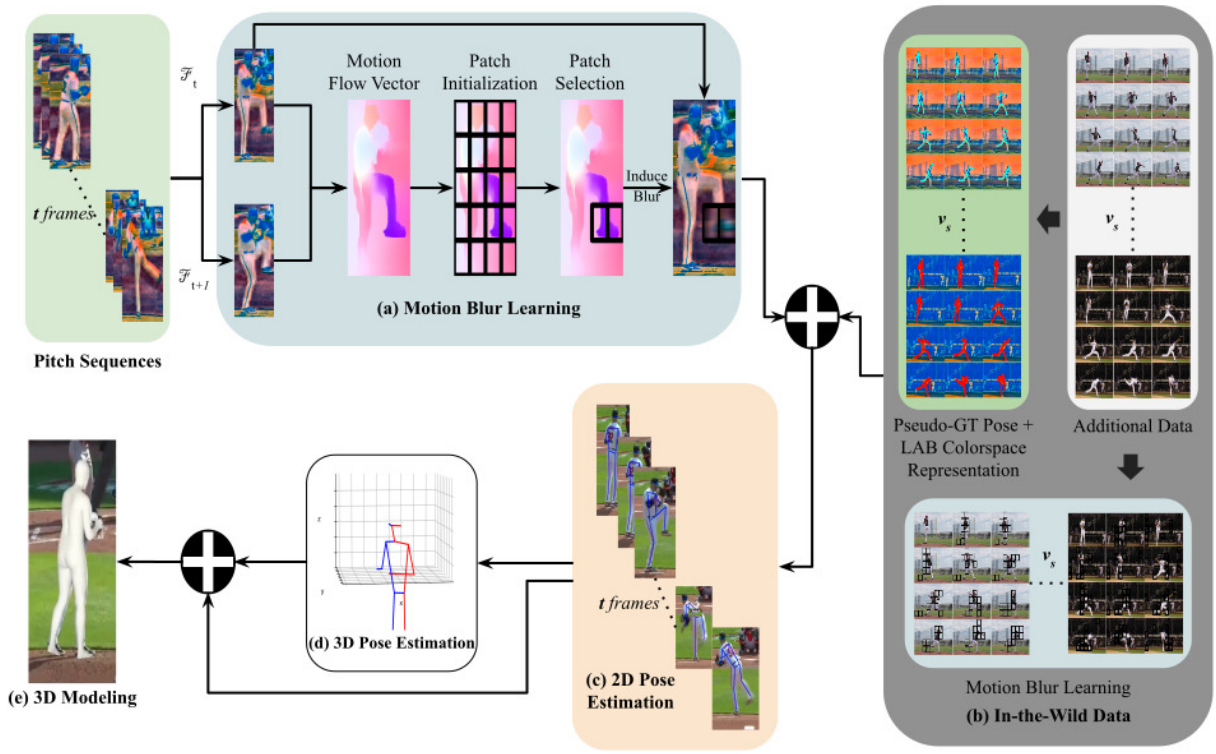


Figure 5.2: Overview of the proposed system. (a) The motion blur learning module creates synthetic blur effects on the pitch sequences to learn better features and generalize well despite such effects. (b) In-the-wild data is leveraged to enhance the robustness of the model on diverse environmental conditions. (c) A regressor-based 2D pose estimator to train the data from the motion blur learning module and the in-the-wild data. (d) A transformer-based 3D pose estimator to train on sequential 2D pose data to estimate the 3D pose. (e) Concatenation of the 2D and 3D poses to estimate the 3D body mesh using spectral GCN.

network to compute the attention matrix based on self-similarities to study the long-range dependencies between pixels of the same reference frame, which is then used to aggregate the motion features represented as shown in Equation (5.1) which is then augmented to the method proposed by Recurrent All-Pairs Field Transforms (RAFT) [113]. This motion flow algorithm was utilized specifically, considering the fact that it can handle occluded regions well since it also considers the self-similarities between frames.

$$\hat{\mathcal{F}}_t = \mathcal{F}_t + \lambda \sum_{j=1}^{N} A(\theta(\S_t), \phi(\S_j), \sigma(y_j)) \tag{5.1}$$

Here, f_i represents motion features, θ, ϕ and σ denotes the projection of the query, key, and value, λ denotes the learned parameter and A denotes the self-similarity attention function.

The procedure for selectively inducing motion blur in specific regions involves a sequential two-step approach following the acquisition of motion flow vectors between consecutive frame pairs. This method comprises Patch Initialization and Patch Selection stages. In the Patch Initialization stage, individual images are partitioned into 4×5 patches. Subsequently, in the Patch Selection stage, a total of $\mathcal{N}$ patches are identified by ranking them based on the magnitude of their associated motion flow vectors. These selected patches are chosen as the targeted regions for inducing motion blur. Notably, this two-step process offers a distinct advantage by ensuring the introduction of motion blur is focused on regions that exhibit significant motion.

The process of applying motion blur to the chosen patches involves the utilization of a motion blur filter. This filter is inherently oriented and is parameterized by a rotation matrix. The angle of rotation (ω), and the scale factor s_f are key determinants of the filter's behavior. This oriented filter is centered at coordinates $(k_s/2, k_s/2)$, where k_s denotes the kernel size. Notably, this filtering approach is applied to the selected $\mathcal{N}$ patches. The filtering procedure can be formally represented as:

$$\mathcal{B}(x, y) = \frac{\mathcal{R}}{\sum \mathcal{R}} * I_{\mathrm{k}}(x, y) \tag{5.2}$$

where,

$$\mathcal{R} = \begin{bmatrix} \cos(\omega)\cdot s_f & -\sin(\omega)\cdot s_f & (k_s//2)\cdot(1-\cos(\omega)\cdot s_f)+(k_s//2)\cdot\sin(\omega)\cdot s_f \\ \sin(\omega)\cdot s_f & \cos(\omega)\cdot s_f & (k_s//2)\cdot(1-\cos(\omega)\cdot s_f)-(k_s//2)\cdot\cos(\omega)\cdot s_f \end{bmatrix} \tag{5.3}$$

Here, $\mathcal{B}(x, y)$ represents the filtered pixel value at position (x,y) of the blurred patch. I_k denotes the k_{th} patch of the image and $(*)$ denotes the convolution operation between rotation matrix $\mathcal{R}$. The summation of $\mathcal{R}$ in the denominator of the fraction ensures normalization, preserving the intensity of the patch. This normalization prevents unwanted intensity changes and contributes to a realistic motion blur effect.

5.3.2 In-the-Wild Video Integration

Furthermore, to enhance the generalizability and robustness of the pose estimator, we leveraged videos from various public sources, featuring slow-motion recordings of pitching actions in professional baseball games or practice sessions, captured at high resolution. This strategic inclusion of external data offered two pivotal advantages: diversity and an abundance of training data, both of which aided to capture a wide spectrum of pitching scenarios, which included different players, camera angles, lighting conditions, and pitching styles.

First, a diverse set of videos (v_s) were captured from publicly available sources on the Internet. These videos consisted of slow-motion videos of pitching action in professional baseball games or practice seasons captured at high resolution. Then, to train our pose estimator using these videos, we first estimated the pose of the pitcher in each frame, effectively generating pseudo-ground truth data for the corresponding frames. Next, to emulate the effects of movements and challenges often observed in low-quality videos, we employed the blurring strategy proposed in Section 5.3.1. This technique induced motion blur in all fast-moving regions of every image within the videos. By subjecting the model to this motion-blurred data, it learned to handle scenarios characterized by motion artifacts and low-quality video conditions, thus enhancing its resilience in practical settings.

Consequently, the motion blur-induced images, along with their corresponding pseudo-ground truth data were included in the training dataset alongside the existing data. This comprehensive training approach capitalized on the combination of diverse video sources, accurate pose estimation from high-quality videos, and exposure to motion blur-induced images, resulting in a pose estimator that demonstrated robustness and proficiency in estimating poses, especially under challenging conditions.

5.3.3 Human Pose Estimation

Estimating the 3D body model of the pitcher offers several distinct advantages over traditional 2D and 3D pose estimation approaches. It facilitates comprehensive analyses,

including the assessment of the pitcher's interaction with the environment, accurate computation of pitch trajectories and release points, and detailed biomechanical evaluation. By harnessing the power of 3D body model estimation, our understanding of the pitcher's movement patterns and biomechanics becomes significantly enriched, leading to valuable insights for optimizing performance and injury prevention strategies.

Thus, to estimate the 3D body model of pitchers in each frame, we first enhance the training data by incorporating synthetic artifacts proposed in Sections 5.3.1 and 5.3.2. This augmented dataset is then used to train a regressor-based 2D pose estimator as described in PEFormer [93]. In contrast to SOTA estimators that rely on heatmaps, we opted for a regressor-based approach due to potential challenges in achieving accurate overlap between the 2D pose of the pitcher obtained through the optimization process of the camera projection (explained in Section 3.2.4). Subsequently, a vision transformer network proposed by MHFormer[68] is used to lift sequences of estimated 2D poses to 3D, resulting in the 3D pose for each corresponding input frame. The loss functions leveraged for 2D and 3D pose estimator are defined as:

$$\mathcal{L}_{pose} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \frac{1}{\mathcal{J}} \sum_{j=1}^{\mathcal{J}} \| kp_{pred}^{(ij)} - kp_{gt}^{(ij)} \|_{\gamma}, \tag{5.4}$$

where,

$$\| \cdot \|_{\gamma} = \begin{cases} \| \cdot \|_2, & \text{if } \gamma = 2 \text{ (for } \mathcal{P}_{2D}) \\ \| \cdot \|_3, & \text{if } \gamma = 3 \text{ (for } \mathcal{P}_{3D}) \end{cases}$$

and $kp_{pred}^{(ij)}$ and $kp_{gt}^{(ij)}$ corresponds to the estimated and ground truth pose from the pose estimator. $\| \cdot \|_{\gamma}$ denotes the Euclidean distance between the γ dimensional pose keypoints.

$\mathcal{P}_{2D}$ and $\mathcal{P}_{3D}$ obtained for each frame are then concatenated into $\mathcal{P}_{concat} \in \mathbb{R}^{\mathcal{J} \times 5}$ and fed into a spectral convolution network [17] inspired by the works of Pose2Mesh [28]. The goal is to directly map the concatenated 2D and 3D poses to the body mesh of the pitcher. The loss function employed to train the mesh network is defined as:

$$\mathcal{L}_{mesh} = \lambda_v \mathcal{L}_v + \lambda_j \mathcal{L}_j + \lambda_n \mathcal{L}_n + \lambda_e \mathcal{L}_e \tag{5.5}$$

Here, $\mathcal{L}_v$ represents the L_1 distance between the output mesh and the ground truth mesh. $\mathcal{L}_j$ measures the loss between the 3D pose of the predicted mesh and the ground truth mesh. $\mathcal{L}_n$ corresponds to the loss of smoothness consistency and $\mathcal{L}_e$ denotes the loss of edge length consistency. The weights for each loss function are denoted by λ.

5.4 Experimentation

Training details. The training process was conducted on a system equipped with an Intel i7 processor (16 cores, 16 GB RAM) and an Nvidia 3050Ti GPU with 4GB of dedicated RAM. For the training process, the dataset was split according to the specifications mentioned in the Chapter 3. Both pose estimator models underwent training for 100 epochs, utilizing a batch size of 16 along with a total of 16 workers for concurrent processing.

For 2D pose estimation, a cross-covariance encoder [3] was employed, along with a simple transformer decoder [65]. The input image was divided into patches of size 16×16, which were then flattened into tokens. The AdamW optimizer was used with a weight decay of 10^{-4}, and the learning rate was set to 10^{-5} for the encoder and 10^{-4} for the decoder. For 3D pose estimation, the Adam optimizer was utilized along with a Reduce Learning Rate on the Plateau scheduler. The scheduler had a patience of 5 and reduced the learning rate by a factor of 0.3. The sequence length was fixed at 27. The mPJPE [45] was employed as a loss function for pose evaluation, as shown in Equation (5.4). GT mesh models were not available for training the 3D models. Thus, pseudo-ground truths were generated using the method described in [95]. The Learning Rate (LR) was initialized as 10^{-4}, and a multistep LR scheduler with LR factor of 0.1 was used. The RMSProp optimizer [43] was used to optimize the model during training.

5.4.1 Motion Blur Learning

A thorough comparative analysis against existing approaches can be visualized from Figure 5.3, which highlights the substantial advancements achieved through our proposed method, particularly evident during pitching actions in the second row. The results demonstrate a notable enhancement in the 3D body model, reaffirming the effectiveness and superiority of our approach.

To strike a balance between augmentation and the complexity of the augmenting task, it is pivotal to avoid over-augmentation, as it can lead to overfitting of the network and hinder performance on unseen data. To determine the optimal hyperparameters, we performed three additional experiments, the results of which are presented in Tables 5.1 - 5.3.

Varying the number of filters

We aimed to find the optimal number of filters that could accurately simulate realistic motion blur effects. By varying the number of filters used, we assessed the performance of

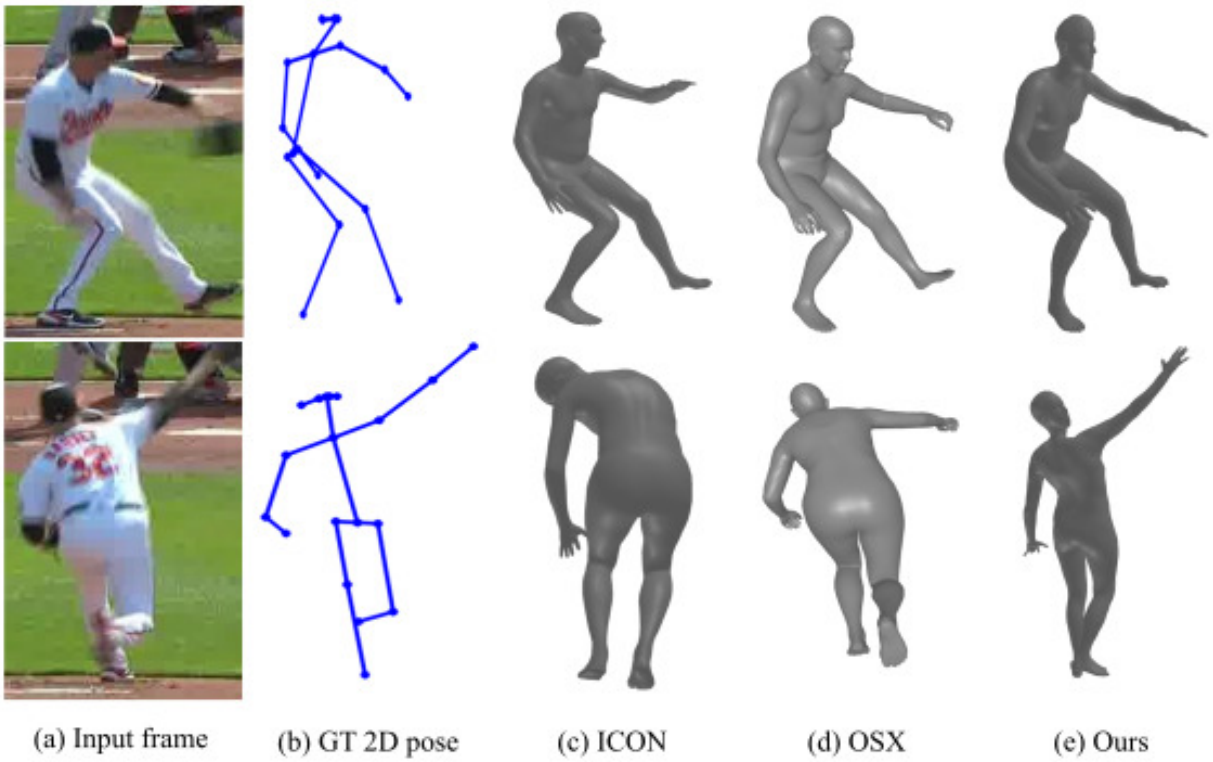


Figure 5.3: Qualitative evaluation of 3D human model in handling motion blur effects.

the pose estimator as shown in Table 5.1.

Table 5.1: Ablation study on varying number of filters for motion blur effect.

Filters	Loss
0	1.15
1	0.68
2	**0.55**
3	1.43
4	2.28
5	3.44

The increasing test loss beyond a certain point in Table 5.1 indicates that the network encounters difficulties in extracting informative features necessary for accurate pose estimation as the degree of motion blur intensifies. This shows that an excessive number of blur filters will lead to a diminishing capacity of the network to effectively handle and interpret motion blur in images.

Different Patch Size

We conducted a study to investigate the impact of different patch sizes on the performance of the pose estimator. By varying the patch sizes (s_{patch}) and the number of patches ($\mathcal{N}$) in the input frames, we assessed how these factors influenced the accuracy of the pose estimator. This experiment enabled us to discover the best s_{patch} and $\mathcal{N}$ for optimal performance.

Table 5.2: Ablation study on the region size and frequency of motion blur effect

s_{patch} \ $\mathcal{N}$	1	3	5	7	9
10	0.83	0.74	0.66	0.64	0.67
20	0.71	0.57	0.62	0.60	0.62
30	0.68	**0.55**	0.61	0.639	0.59
40	0.74	0.63	0.68	0.75	0.78
50	0.77	0.75	0.71	0.83	0.97

The extensive study conducted in 5.2 demonstrates that the optimal results were achieved when using three patches, each with a patch size of 30. This finding aligns with qualitative observations, as this particular hyperparameter setup resulted in the most realistic blur pattern.

Different Patch Types

To identify the most suitable patch type that closely resembles a realistic representation of the motion blur effect, we conducted experiments involving different patch types. The results of this experiment are summarized in Table 5.3.

Table 5.3: Comparison with different patch types

Patch Type	Loss
None	1.15
Binary Mask	2.12
Inpainting	1.57
Gaussian Blur	0.99
Motion Blur	**0.55**

As shown in the results in Table 5.3, motion blur filters emerged as the superior method by successfully capturing the essence of rapid motion in a more accurate manner.

Taking into account the findings from Table 5.1 to 5.3, the most optimal setup was with the adoption of a motion blur patch type that utilizes 2 filters, a patch size of 30, and 3 patches. This combination of parameters has demonstrated a greater ability to provide a substantial representation of the data, thereby significantly enhancing the generalizability of the pose estimator in handling motion blur effects. This finding contributes to effective pose estimation by handling challenging scenarios characterized by fast-moving actions in the images.

5.4.2 Human Pose Estimation

Different data modules

To evaluate the effectiveness of our method, we conducted tests on the curated dataset as described in Chapter 3. Specifically, we evaluated the performance of our 2D and 3D pose

estimation algorithms after augmenting the dataset with motion blur effects and In-the-Wild (ItW) videos. The results of these evaluations are summarized in Table 5.4.

Table 5.4: Results of the estimated pose with different modules for training.

Base Model	ItW	Blur	2D Loss	3D Loss
✓			1.05	1.93
✓	✓		0.88	1.61
✓		✓	0.55	1.47
✓	✓	✓	**0.48**	**1.23**

By incorporating both ItW data and motion blur modules, the performance of the base model is significantly improved. The 2D loss shows a substantial improvement of 58%, indicating enhanced accuracy in estimating the 2D pose of the human body. Subsequently, this improves the performance of the 3D pose estimator by 36%.

Comparison on SOTA pose estimators

The experimental evaluation in Table 5.5 demonstrates the performance improvement achieved by incorporating our approach during the training of SOTA 2D pose estimators using our dataset. Our objective was to show the efficacy of our approach in improving the overall performance of the pose estimators.

Table 5.5: Performance of different SOTA 2D pose estimation approaches with the proposed motion blur learning module.

Method	Type	Motion Blur	Loss
Xu et al [125]	Heatmap		1.37
Ke et al [111]	Heatmap		1.46
Panteleris et al[93]	Regressor		1.15
Li et al. [66]	Heatmap		1.83
Mao et al. [85]	Regression		1.26
Xu et al [125]	Heatmap	✓	1.17 (+0.20)
Ke et al [111]	Heatmap	✓	1.21 (+0.25)
Panteleris et al[93]	Regressor	✓	0.55 (+0.60)
Li et al. [66]	Heatmap	✓	1.46 (+0.37)
Mao et al. [85]	Regressor	✓	0.61 (+0.65)

The results indicate a significant improvement in the pose estimators' performance after integrating with the proposed approach, primarily due to its ability to handle motion blur. Furthermore, the comparison between heatmap-based and regression-based techniques highlights the limitations of the former in addressing the challenges of our dataset discussed in Section 5.3.3.

5.5 Summary

The chapter proposes a unique approach to accurately estimate the pose of pitchers in baseball games by addressing the challenges posed by the motion blur effect. An innovative augmentation technique has been proposed to increase the frequency and consistency of motion blur in a strategic pattern to enhance the network's ability to learn and adapt to these effects. Integrating in-the-wild video data into the training module with psuedo-ground truth pose information aided the network to be effective against the variable lighting and camera positions.

By training a 2D and 3D pose estimator on these data, significant improvements in the accuracy of pose estimation have been observed, particularly during pitching actions. Thus, this approach demonstrates a more focused and strategic augmentation strategy to induce motion blur can yield improvements in pose estimation emphasizing the importance

of thoughtful augmentation in addressing the motion blur effect and offering an alternative perspective to traditional complex pipelines.

Chapter 6

PitcherNet: Powering the Moneyball Evolution in Baseball Analytics

6.1 Overview

Current research on baseball game analysis often relies on numerical databases containing pre-recorded offline data [13, 42, 108, 122]. These methods typically focus on predicting actions or game statistics based on these historical records. While some approaches use real-time data, they are often limited to controlled laboratory environments with expensive motion capture setups [84, 92, 103]. This restricts the generalizability of their findings to the dynamic and complex situations encountered during live games. Live game broadcasts, however, offer a more holistic perspective by capturing the entirety of a pitcher's motion within the game's natural environment. This approach overcomes the limitations of controlled settings. However, analyzing broadcast data presents its own challenges, such as motion blur and low video resolution, which can significantly hinder accurate pitch analysis and potentially lead to unreliable results.

To bridge the limitations of existing methods and address the challenges of analyzing real-world live broadcasts, we introduce PitcherNet, an end-to-end automated system designed to predict performance-critical pitch statistics from the kinematic data derived from live broadcast videos. PitcherNet transcends existing approaches by meticulously analyzing each stage of the pitcher's movement, from player identification to pose estimation, and finally pitch analysis. Some crucial pitch statistics that PitcherNet estimates include pitch position, pitch velocity, ball release point, and release extension. Human mesh recovery and pitch statistics derived from the PitcherNet system are illustrated in Figure 6.1. To

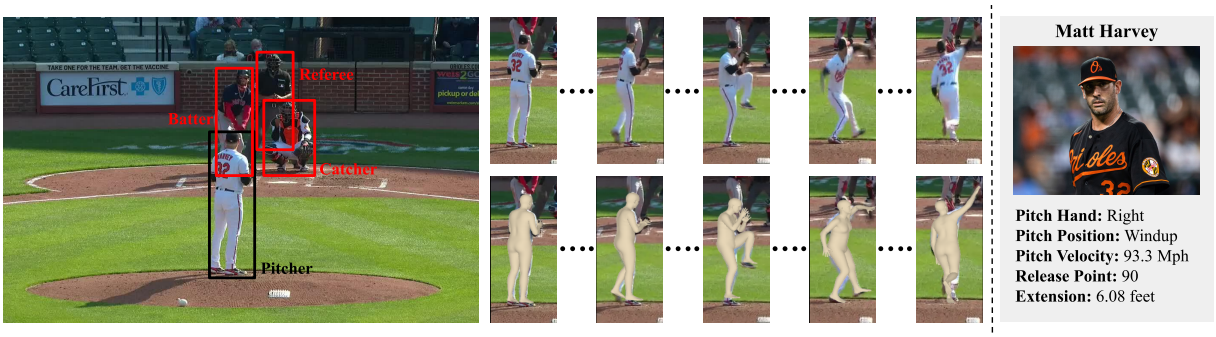


Figure 6.1: **3D player reconstruction and kinematic-driven pitch statistics from monocular video.** We introduce *PitcherNet*, a pioneering deep learning system that tackles low-resolution video limitations through efficient 3D human modeling for robust player alignment (left) and reliable pitch statistics analysis from estimated kinematic data (right).

the best of our knowledge, this is the only system, that extracts pitch statistics extensively driven from the pitcher kinematics from low-quality broadcast videos.

6.2 Preliminary

6.2.1 Player Tracking and Identification

Various approaches for player identification exist, primarily relying on either facial features or jersey numbers. In works such as [83, 6], facial recognition is employed to label players based on detected face regions. Conversely, jersey number recognition, as seen in [5, 132, 76, 37, 116], is a prevalent method of player identification. Vats *et al.* [116] recently introduced a comprehensive offline tracking framework for ice hockey, employing 1D convolutions for team and jersey number identification. Sentioscope [8] tracks player interactions using a dual camera setup and model field particles on a calibrated soccer field plane. DeepPlayer [132] proposes a multicamera player identification system integrating jersey number patterns, team classification, and pose-guided partial features. Works such as [21, 5] incorporate end-to-end trainable spatio-temporal networks for identifying jersey numbers in ice hockey and soccer. Additionally, [104, 75, 132] utilize convolutions to extract features and exploit information from the pose of the players to determine the numbers of the jersey.

Existing player identification approaches rely heavily on distinctive features (such as clothing, jersey number, and facial features). However, these features are often unreliable due to clothing variations, occlusions, and varying camera angles. To address these challenges, we propose a novel approach that decouples player actions from individual tracklets. This approach shifts the focus from specific player features to the action itself, enabling robust and accurate player identification even when traditional features fail.

6.2.2 Baseball Pitch Statistics

Previous research has mostly focused on estimating the pitch statistics from existing baseball data collection database such as the PITCHf/x system [13, 108, 122]. Works such as [13] leveraged these prior game statistics to classify pitch types using Support Vector Machines (SVM) with linear kernel functions, and [108] utilized Linear Discriminant Analysis (LDA), decision trees, and SVM to find the best apt model to classify pitch types. Hickey *et al.* [42] aimed to improve the interpretability alongside accuracy of classification models used for pitch prediction.

Recently, Manzi *et al.* [84] proposed a descriptive laboratory study in the setting of 3D motion-capture to classify pitch throws by analyzing pitcher kinematics. Similarly, Oyama *et al.* [92] used motion capture systems to validate the pitching motion of the baseball by comparing with the calculated angles. Chen *et al.* [23] proposed a network which can recognize hand pitching style (overhand, three-quarter, etc.) by extracting the human body segment and a descriptor representation using star skeletons.

6.3 Methodology

The overview of the proposed system, PitcherNet, is presented in Figure 6.2. The system is divided into three components: (1) **Player Tracking and Identification**, involving the initial detection of all players, subsequent tracking of detected players with the assignment of unique labels to each tracklet, and the *decoupling of actions* from the inferred poses of the players in each sequence in the tracklet to facilitate player classification; (2) **3D Human Modeling** utlizes a 3D human model prior [79] to estimate the pose of the player guided by masked modeling, distribution learning, and silhouette masks; and (3) **Pitch Statistics** leveraging Temporal Convolutional Network (TCN) and kinematic-driven heuristics to reliably capture various pitch metrics. This section provides a comprehensive exploration of each component, elucidating the underlying design choices aimed at enhancing the performance of existing techniques in the context of the proposed system.

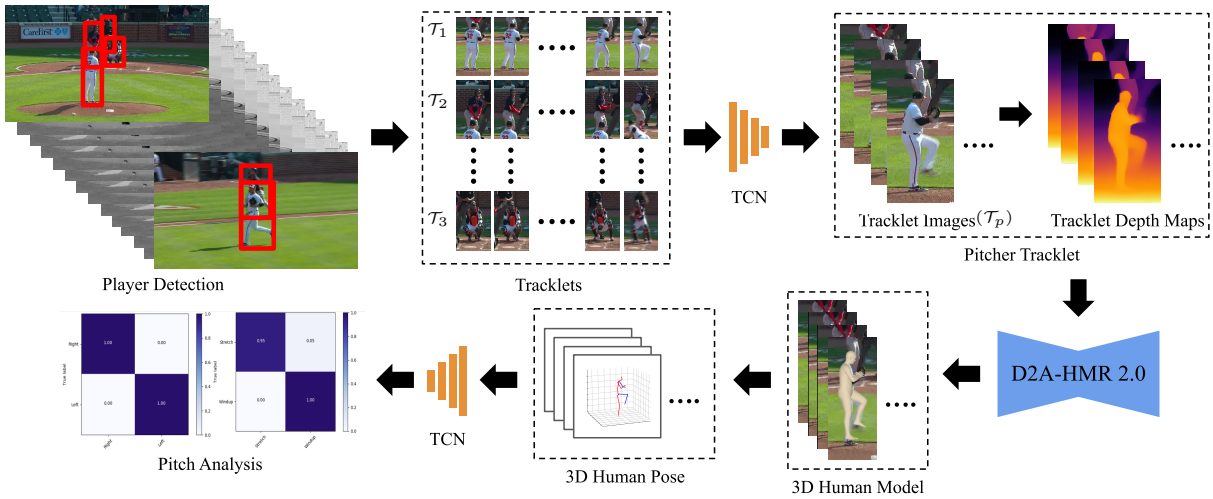
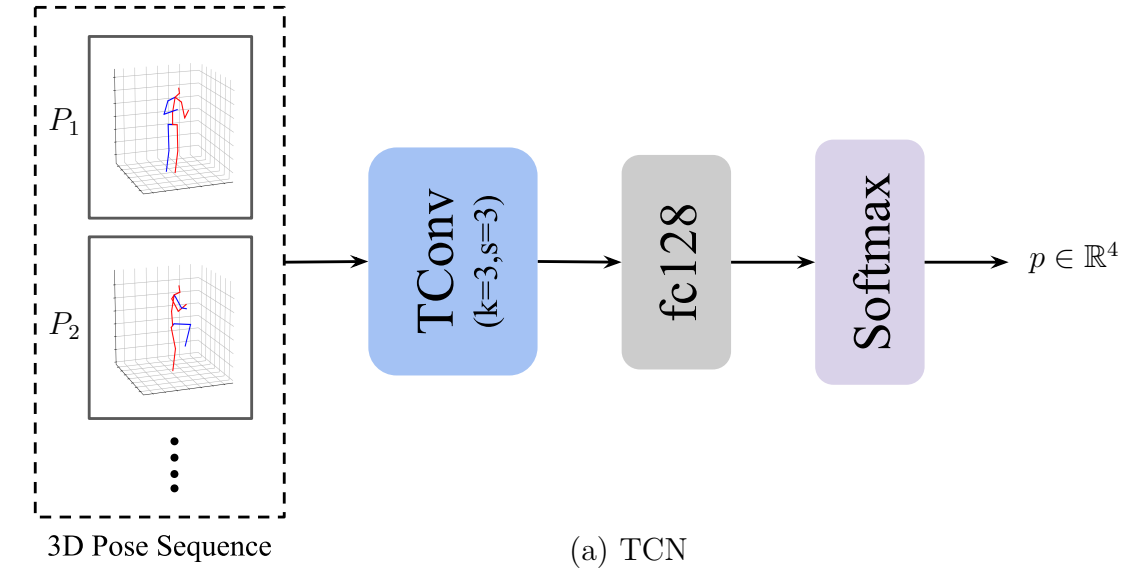


Figure 6.2: **Overall architecture**. Given a video broadcast, we begin by extracting player tracklets, denoted as $\mathcal{T} \in \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n\}$. Each tracklet $\mathcal{T}_k$ consists of a sequence of frames $\mathbf{F}_i$ where $\mathbf{F}_i \in \mathbb{R}^{H \times W \times 3}$ for N frames. These tracklets are then processed through a TCN, which implicitly decouples player actions and identifies the tracklet of the pitcher, called $\mathcal{T}_p$. Subsequently, $\mathcal{T}_p$ undergoes encoding via an encoder ($\mathbf{E}$) to derive pseudo-depth information for each frame. The frames, along with their corresponding pseudo-depth data, are fed into a 3D modeling technique (D2A-HMR 2.0). This framework is responsible for predicting the 3D mesh and 3D joint positions of the pitcher, facilitating detailed analysis of various pitch metrics using the temporal kinematic information processing the 3D joint positions.

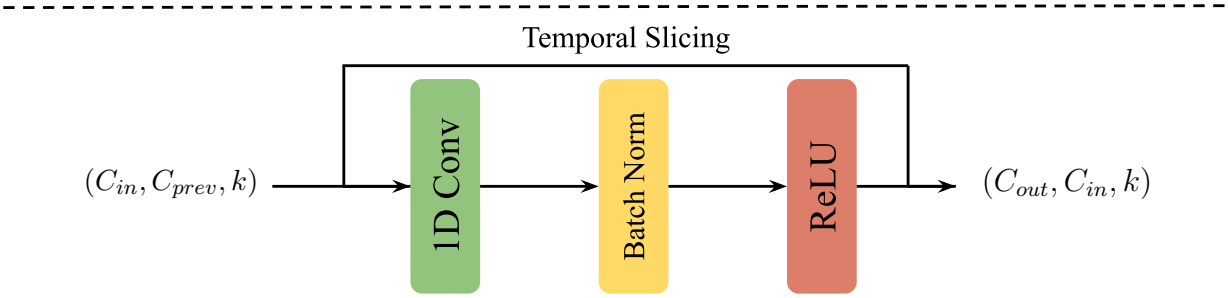
6.3.1 Player Tracking and Identification

Accurate tracking and identification of players are fundamental for effective action recognition and analysis in sports scenarios. As highlighted in the literature, the challenges associated with simultaneous tracking and classification based on features are the compromise in the reliability of obtaining the desired tracklet. Thus, our objective is to *decouple the action from kinematics obtained from sequences of the tracklets* to acquire the desired tracklet ID for subsequent downstream tasks.

Initially, tracklets are generated using the methodology proposed in SORT [9], which utilize YOLOX [36] detections. Each tracklet is assigned a unique identifier along with the 3D pseudo-pose of MHFormer [68]. Subsequently, we decouple player actions by classifying each tracklet into the player's role (pitcher, batter, or others). Given that pitchers are the



(a) TCN



(b) TConv

Figure 6.3: **Temporal Convolutional Network. (a)** Overview of the proposed TCN for the player identification task, where fc denotes fully connected layers and p refers to the model's output. **(b)** Architecture of the TConv block used in the TCN, where C_{in}, C_{out} and C_{prev} denotes the input, output and previous channels, respectively and k denotes the kernel size.

primary focus of our investigation, we identify sequences within tracklets where pitching actions occur. To accomplish this, we employ a TCN architecture designed to decouple various actions within each tracklet, specifically isolating the pitching action of interest. The TCN architecture, described in Figure 6.3, eliminates the dependence on the characteristics of specific players for classification, providing a more robust solution to identify the target player in dynamic sports scenarios.

The TCN architecture utilizes a series of five TConv layers which encompass a dilated 1D convolutional layer with progressively increasing dilation rates, followed by batch normalization and ReLU activation in each layer. The network ingests a 4D tensor representing pose sequences $\mathbb{P} = \{P_i : P \in \mathbb{R}^{K \times C}\}_{i=0}^{N}$, where each dimension corresponds to batch size (B), temporal sequence length (N), number of joint positions (K), and 3D player coordinates (C). This progressive dilation allows the TCN to capture long-range temporal dependencies crucial for understanding complex motion patterns, while dropout layers and batch normalization enhance the model's generalizability. In addition, skip connections are utilized, allowing the model to directly access information from the original input at a deeper stage in the network. This helps to address the problem of vanishing gradients and improve the flow of information throughout the TCN architecture.

6.3.2 3D Human Modeling

Estimating the pose of the pitcher is crucial for effective pitch analysis of the players from live broadcast video. The input to the 3D human modeling technique is the player of interest tracklet from the player tracking and identification component (Section 6.3.1). To enhance the reliability of pose estimation in challenging, real-world scenarios, we have advanced the D2A-HMR technique introduced in [15] specifically focusing on our input data conditions.

D2A-HMR focuses on learning the underlying output distribution with the objective of minimizing the distribution gap. The method takes an input image and pseudo-depth maps, utilizing a transformer encoder that incorporates cross- and self-attention mechanisms along with learnable gate fusions to produce the final output token. Subsequently, a decoder is employed to predict the person's silhouette, guiding the overall structure of the player in the input image. Additionally, a regression head is utilized to obtain the mesh vertices. In this work, we have introduced several design enhancements to augment D2A-HMR model, and these modifications will be referred to as D2A-HMR 2.0 modeling technique.

Algorithm 1 outlines the core steps of the D2A-HMR model. Initially, a depth encoder

Algorithm 1 Distribution and Depth Aware Human Mesh Recovery

Input: Image ($\mathbf{I}$)
Initialization:
$\boldsymbol{E}(\mathbf{I}) \rightarrow \mathbf{D}$
$\boldsymbol{F}(\mathbf{I}, \mathbf{D})$
Positional Embedding:
$P_e(= \omega_1 P_l + \omega_2 P_s) \rightarrow z_{\text{img}}, z_{\text{depth}}$
Self-Attention (MHSA):
$\text{MHSA}(z_{\text{img}}) \rightarrow z'_{\text{img}}$
$\text{MHSA}(z_{\text{depth}}) \rightarrow z'_{\text{depth}}$
Cross-Attention (MHCA):
$\text{MHCA}(z'_{\text{img}}, z'_{\text{depth}}) \rightarrow z_{\text{c}}$
Learnable Fusion Gates:
$z = \omega_3 z'_{\text{img}} + \omega_4 z'_{\text{depth}} + (1 - \omega_3 - \omega_4) z_{\text{c}}$
Masked Modeling:
$q_{mask} = \text{Mask}(z)$
Distribution Matching:
$\boldsymbol{R}(z) \rightarrow \sigma, \mu$
$\bar{\mu} = (\mu - \mu_{gt})/\sigma \rightarrow NF \rightarrow \mathcal{L}_{RLE}$
Silhouette Decoder:
$\mathbf{I}_{silh} = \boldsymbol{D}(z, k, s, p)$
Output: 3D mesh vertices, $\mathcal{P} = \boldsymbol{R}(z)$, $\mathcal{P} \in \Re^{6890 \times 3}$

$E(I)$ takes an input image (I) and generates a depth map (D). Concurrently, both I and D are fed as input to the feature extractor (F), followed by hybrid positional encoding P_e, yielding tokens z_{img} and z_{depth}. These tokens subsequently undergo processing by self-attention and cross-attention modules, resulting in z'_{img}, z'_{depth} and z_c respectively. Fusion gates then merge these outputs into a singular token, z. Refer to Chapter 4 for information for D2A-HMR architecture design and experimental results.

D2A-HMR 2.0 leverage a depth encoder called Depth Anything [126] to extract pseudo-depth, which utilizes the DINOv2 encoder [91] and the DPT decoder [98]. In addition to the regression of the mesh vertices as output, we also extract the 3D joint coordinates (J_{3D}). Following [48, 55, 28], the mesh vertices are further regressed to find the 3D regressed joint coordinates (J^r_{3D}) using a predefined regression matrix $G \in \mathbb{R}^{K\times M}$. Here K and M correspond to the number of joint positions and the number of vertices. Then, the final 3D pose $(\hat{J}_{3D})$ of the person in the input image is formulated as shown in Equation (6.1).

$$\hat{J}_{3D} = \omega_1 J_{3D} + \omega_2 G V_{3D} = \omega_1 J_{3D} + \omega_2 J^r_{3D} \tag{6.1}$$

where ω_1 and ω_2 are weights for the joint distribution. V_{3D} denotes the vertices of the output mesh.

To further enhance the efficacy of D2A-HMR 2.0, we integrate a substantial volume of unlabeled data sourced from the Internet to facilitate robust generalization in dynamic real-world scenarios, as illustrated in Figure 6.4. This augmentation involved the implementation of a transformer network with pretrained weights initialized using MHFormer [68]. Specifically, we select high-resolution practice videos from the Internet and introduce motion blur artifacts. Prior to inducing blur, we use a transformer network to predict its corresponding 3D poses, leveraging its superior performance with high-resolution data.

6.3.3 Pitch Statistics

The pitch type, a complex interplay of various pitch statistics, ultimately determines the pitch delivered. This work focuses on estimating crucial kinematically derived pitch statistics such as pitch position, release point, release extension, pitch velocity, and handedness from the 3D pose data obtained using the D2A-HMR 2.0 model. While factors such as break and spin rate also influence pitch action [89], this work focuses on these core kinematic statistics mentioned above. By analyzing these statistics, we gain valuable insight into the mechanics of pitch delivery. These pitch statistics combined with kinematic motion

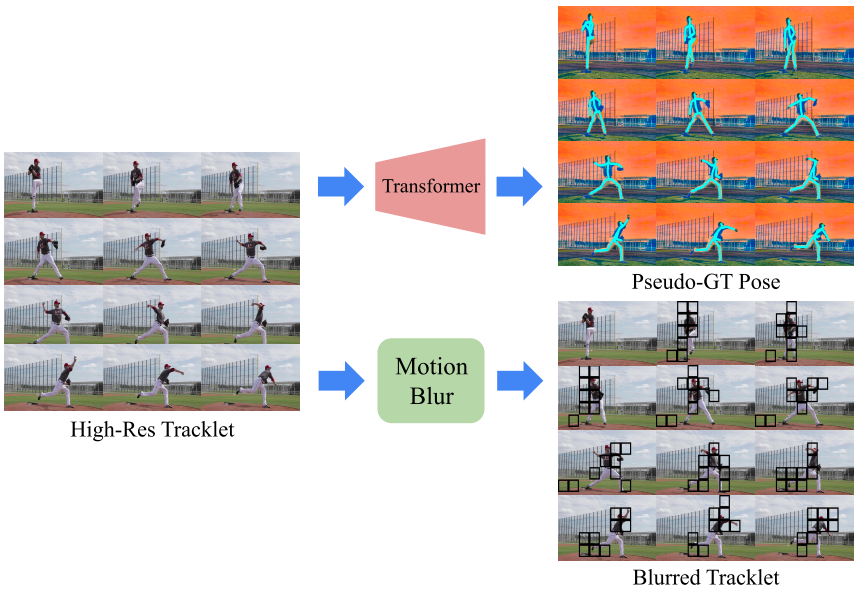


Figure 6.4: **Data Augmentation Technique.** Pseudo-ground truth pose is collected using a Transformer model for improved generalizability of the pose estimation model.

data will contribute significantly in the prediction of complex pitch actions. The 3D kinematic information is fed as input to the pitch statistics component to estimate the different pitch statistics including pitch position, release point, pitch velocity, release extension, and handedness.

Pitch Position

Pitchers utilize two legal pitch position styles: the windup, a full-body motion maximizing power, and the set/stretch, a quicker, more compact motion sacrificing some velocity for faster delivery. Mastering these positions allows pitchers to deceive batters by disrupting their timing and pitch recognition [103]. This work employs a TCN backbone, identical to the player identification network, for pitch position classification using a sigmoid activation function. Each video tracklet is fed into the TCN with a sequence length of 100 frames for classification.

Handedness

Accurate determination of the pitcher's handedness is critical for effective pitch analysis. By isolating the throwing hand within each video frame, we can tailor feature extraction to the pitcher's specific mechanics. This work utilizes a TCN to estimate handedness. While the TCN demonstrates effectiveness, simpler methods based on hand appearance analysis might also be suitable for handedness classification. Regardless of the chosen technique, identifying handedness allows the analysis pipeline to account for the pitcher's mechanics, leading to improved feature extraction and ultimately, more accurate pitch statistics.

Release Point

The release point, defined as the specific location where the pitcher releases the ball from their hand toward the batter, plays a crucial role in determining both the pitch velocity and the release extension [121]. It also plays a crucial factor in deciphering tunneled pitchers.

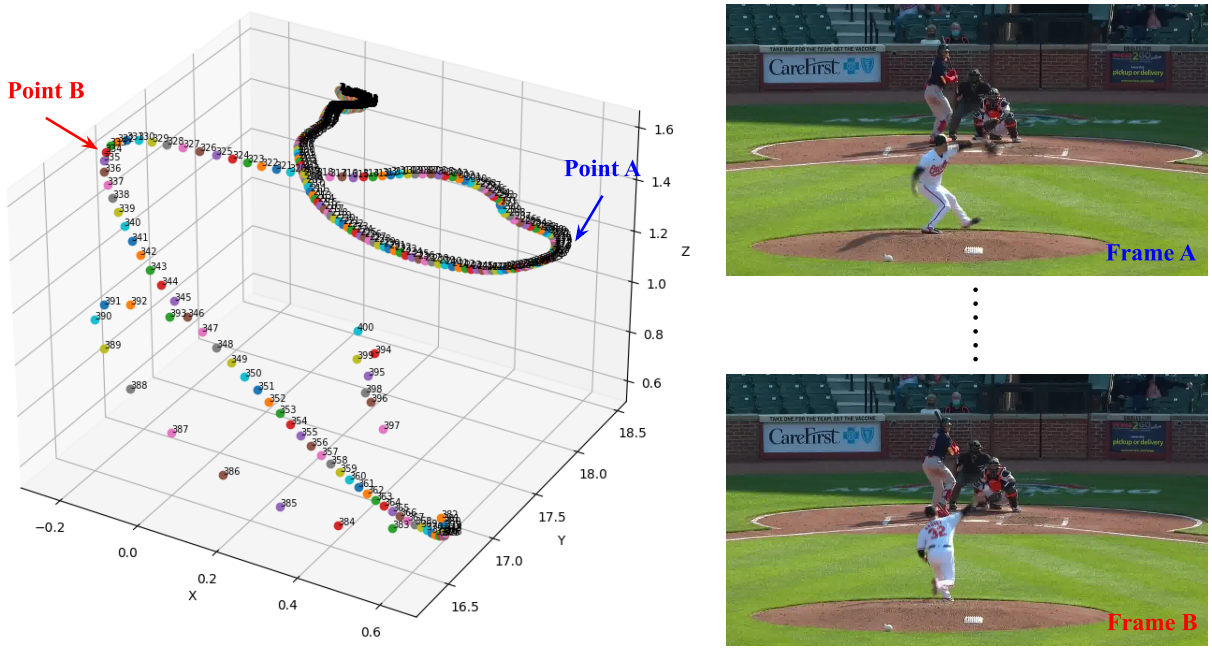


Figure 6.5: **Trajectory of the right wrist joint in 3D space.** Illustration of two frames which correspond to the points (A and B) marked in the trajectory plot that determines the release point.

As illustrated in Figure 6.5, we use the wrist kinematics of the pitcher in the x-plane (lateral movement) to identify the release point. Here, we determine the extreme coordinates in the x-plane to establish the maximum and minimum limits of wrist movement during the throwing motion. The limit point, Point A corresponds to the final cocking phase which refers to the point of maximum external rotation of the throwing shoulder from the glove, while the limit point Point B, represents the end of the acceleration phase with the ball release and start of the follow-through phase [88]. We hypothesis that the ball release point will be one amongst n frames windowed on point B with the peak pitch velocity.

Pitch Velocity

Pitch velocity, measured in miles per hour, reflects the ball's speed upon leaving the pitcher's hand [35]. This study proposes a method to estimate pitch velocity by analyzing changes in the throwing hand's wrist position at the release point identified using the estimated 3D pose data. Equation (6.2) calculates the angular velocity of the wrist at release based on the change in arctangent of consecutive wrist coordinates (w_x, w_y) before (frame $r-1$) and at (frame r) the release point. This angular velocity is then converted to an approximate pitch velocity (v_p) by multiplying it by the lever arm length (l) between the wrist and the elbow joint.

$$v_p = \omega \times l = \{(atan(w_y^r, w_x^r) - atan(w_y^{r-1}, w_x^{r-1})) \times T\} \times l \tag{6.2}$$

As mentioned above in Section 6.3.3, we estimate the ball release point by finding the maximum velocity in a window around Point B using Equation (6.2) which will compute the pitch velocity.

Release Extension

Release extension, in baseball, refers to the distance a pitcher creates between the pitching mound and the release point of the ball towards the batter. The release extension helps to differentiate pitch types, as fastballs typically involve greater extension compared to breaking balls. It essentially describes how much closer the pitcher gets to the home plate at the moment of release compared to where they started their throwing motion. The release extension is mathematically indicated as shown in Equation (6.3).

$$\text{Extension} = \sqrt{(w_x - a_x)^2 + (w_y - a_y)^2 + (w_z - a_z)^2} \tag{6.3}$$

Here, in Equation (6.3), a refers to the position of the pitching leg's ankle joint. The pitching ankle joint is chosen since it remains planted on the mound during the pitch set and the initial part of the delivery.

6.3.4 Loss Functions

Most pitchers tend to be right-handed. Therefore, there is an imbalance in the class, especially in the handedness data of the pitchers. We use focal loss as a loss function to address the issue of class imbalance [72]. More weight is given to minority classes while training using a gamma tuning parameter (set to 2 initially). The loss function used for the estimation of pitch position and handedness is denoted as shown in Equation (6.4).

$$L(p_t) = -\alpha_t * (1 - p_t)^{\gamma} * \log(p_t) \tag{6.4}$$

where α_t and γ are the balancing parameter and sampling focus parameter, respectively. p_t denotes the predicted probability of the true class. The D2A-HMR human model is trained using the objective mentioned in Equation (6.5).

$$\begin{aligned} \mathcal{L}_{model} = \lambda_{RLE}\mathcal{L}_{RLE} + \lambda_{SMPL}\mathcal{L}_{SMPL} + \lambda_{3D}\mathcal{L}_{3D} \\ + \lambda_{2D}\mathcal{L}_{2D} + \lambda_{silh}\mathcal{L}_{silh} \end{aligned} \tag{6.5}$$

where $\mathcal{L}_{RLE}$, $\mathcal{L}_{SMPL}$, $\mathcal{L}_{3D}$, $\mathcal{L}_{2D}$ and $\mathcal{L}_{silh}$ correspond to residual likelihood loss, 3D vertex loss, regressed 3D loss, reprojected 2D loss and silhouette loss. All λ correspond to the weights assigned to distribute the importance of each objective. We incorporated an additional loss function into our D2A-HMR 2.0 as shown in Equation (6.6).

$$\hat{\mathcal{L}}_{model} = \mathcal{L}_{model} + \lambda^{r}_{3D}\mathcal{L}^{r}_{3D} \tag{6.6}$$

Here, $\mathcal{L}^{r}_{3D}$ corresponds to the 3D output joints of the regression head of the D2A-HMR model.

6.4 Experimentation

Implementation Details. The training process is conducted on three NVIDIA A6000 GPUs with 48GB RAM. Adam optimizer with a batch size of 48 with 500 epochs is used

to train the 3D human model. A learning rate of 10^{-4} with betas of 0.9 and 0.99 is used for optimization. The TCN model for handedness estimation and pitch position estimation is trained for 50 and 100 epochs, respectively using one of the three GPUs. The TCN model trained for player identification was trained for 200 epochs using two GPUs with AdamW optimizer with a learning rate of 10^{-2}.

6.4.1 Pitcher Identification

The impact of the pitcher identification task is compared with two temporal networks (LSTM, transformer with only self-attention blocks) in Table 6.1. Simple baseline temporal networks were used for comparison to validate the effectiveness of pose-based role classification. Since these are tasks that use distinct player kinematics, complex networks were not needed.

Table 6.1: Comparison of our model with baseline temporal networks on MLBPitchDB dataset [15].

	Test Accuracy ↑
LSTM	85.55
Transformer	91.11
Ours	**96.66**

Our approach achieves superior test accuracy compared to both LSTMs and transformers with self-attention blocks, as shown in Table 6.1. Specifically, we observe an improvement of 11.11% and 5.55% in accuracy relative to LSTMs and transformers, respectively.

6.4.2 3D Human Modeling

Depth Encoder. Experimentation with different depth encoders including AdaBin [10], ZoeDepth [11], DINOv1 [20], DINOv2 [91] and Depth Anything [126] is done in Table 6.2. The D2A-HMR model proposed in [15] utilizes DINOv2 [91] as its depth encoder for human body modeling.

Table 6.2: Impact on different depth encoders for D2A-HMR evaluated on 3DPW dataset.

	mPJPE ↓	PA-mPJPE ↓
Bhat *et al.* [10]	90.3	55.4
Bhat *et al.* [11]	87.8	53.3
Caron *et al.* [20]	83.1	50.6
Oquab *et al.* [91]	80.5	48.4
Yang *et al.* [126]	**78.7**	**46.9**

Our findings demonstrate that employing Depth Anything [126] as the depth encoder to generate pseudo-depth leads to improvements of 1.8mm and 1.5mm in mPJPE and PA-mPJPE, respectively. This can be attributed to the utilization of a significantly larger dataset of unlabeled images allowing the model to learn more comprehensive visual representations. Qualitative comparison of the various monocular depth estimation techniques on the MLBPitchDB dataset is illustrated in Figure 6.6.

Figure 6.6: **Qualitative results.** Qualitative comparison of the various depth estimation techniques in MLBPitchDB baseball dataset.

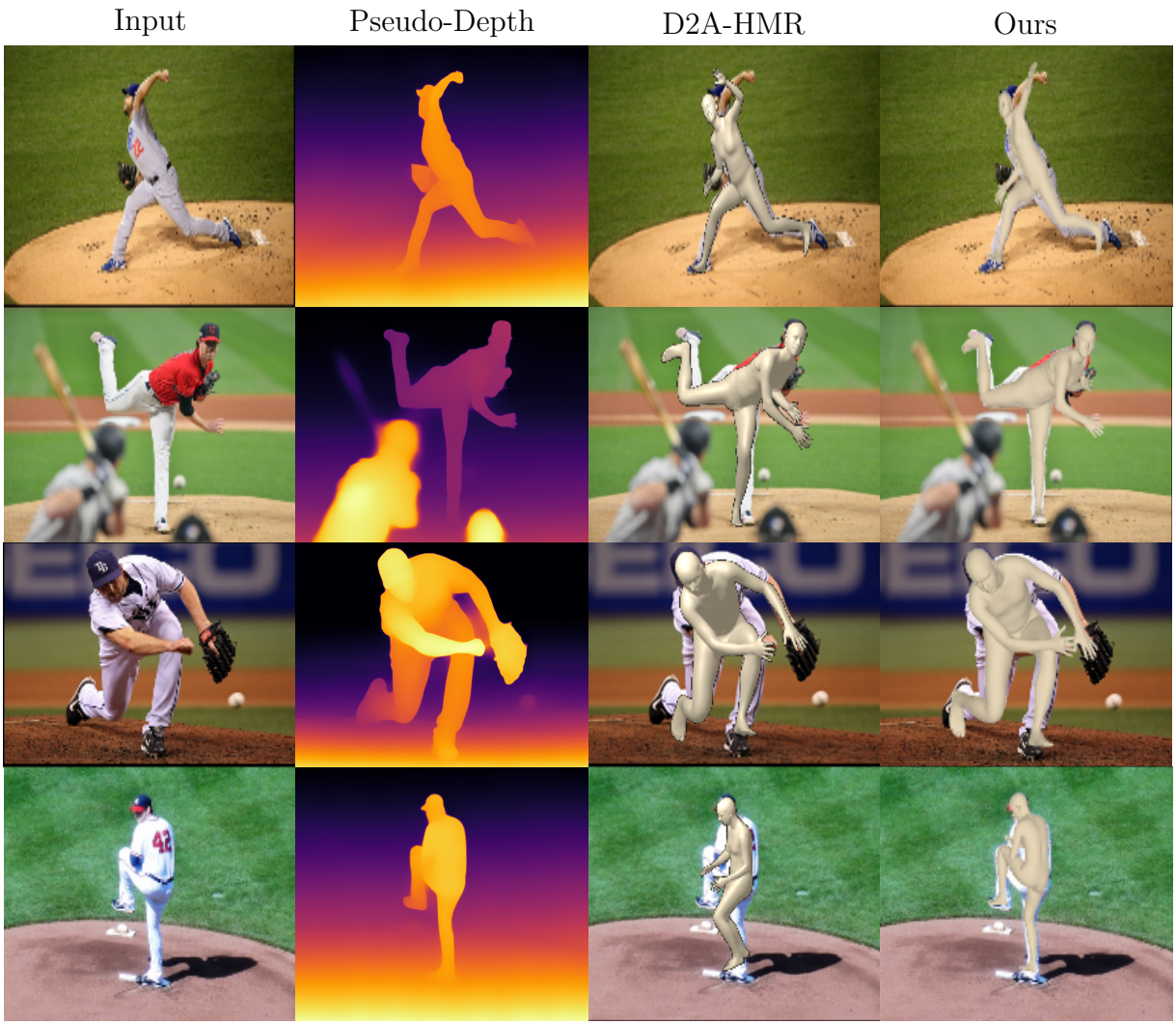


Figure 6.7: **Qualitative results.** Qualitative comparison of the pitcher's mesh alignment with the input image using our D2A-HMR 2.0 model.

Regression Heads. We evaluate the performance of two distinct regressor head architectures within our model. The first design directly regresses the vertex coordinates of the transformer output tokens. In contrast, the second approach predicts both the vertices and the 3D pose of the players. A detailed comparison of their performance is presented in Table 6.3.

Table 6.4: Performance of our pitch statistics module on different pitch metrics including pitch handedness, pitch position, release point, pitch velocity, and release extension in the test dataset compared against baseline temporal networks.

(a) Handedness

	Acc. ↑	F1 ↑	Prec. ↑
LSTM	085.0	085.7	090.0
Ours (TCN)	**100.0**	**100.0**	**100.0**

(b) Pitch Position

	Acc. ↑	F1 ↑	Prec. ↑
LSTM	81.3	82.5	85.0
Ours (TCN)	**97.5**	**97.4**	**95.0**

(c) Release Point

	A_1 ↑	A_2 ↑	A_5 ↑
LSTM	31.3	46.4	63.5
TCN	43.4	51.5	77.6
Ours	**80.8**	**85.8**	**97.9**

(d) Pitch Velocity

	$A_{1\%}$ ↑	$A_{2\%}$ ↑	$A_{5\%}$ ↑
LSTM	05.0	13.1	22.2
TCN	10.1	18.1	48.4
Ours	**43.4**	**68.6**	**94.9**

(e) Release Extension

	$A_{5\%}$ ↑	$A_{8\%}$ ↑	$A_{10\%}$ ↑
LSTM	04.0	07.0	11.1
TCN	14.1	19.1	25.2
Ours	**24.2**	**31.3**	**37.3**

Table 6.3: Ablation study on different regressor heads for D2A-HMR evaluated on 3DPW dataset.

	mPJPE ↓	PA-mPJPE ↓
w/ vertex only	80.5	48.4
w/ vertex+joints	**78.7**	**46.9**

As shown in Table 6.3, our model demonstrates superior performance when incorporating both the player's vertices and 3D joints during the regression process. This is likely due to the additional information provided by the joints, which helps the model refine the predicted 3D pose and achieve more accurate alignment. Furthermore, the model optimizes the output 3D joints by minimizing the difference between them and the ground truth 3D poses, further contributing to the overall improvement in performance. Figure 6.7 demonstrates the superior alignment ability of D2A-HMR 2.0 from a given input image when compared with existing SOTA HMR techniques.

Pseudo-GT Data. The impact of leveraging additional pseudo-ground truth pose data on the training process of the D2A-HMR 2.0 model is showcased in Table 6.5. Specifically, pseudo 2D and 3D pose data obtained from HRNet [111] and MHFormer [68], respectively,

are used as ground truth for training the HMR model.

Table 6.5: Ablation study on utilizing Pseudo-GT data.

	mPJPE ↓	PA-mPJPE ↓
w/o Pseudo-GT data	79.1	47.4
w/ Pseudo-GT data	**78.7**	**46.9**

Our D2A-HMR 2.0 model demonstrates an improvement in performance of recovering the 3D human mesh by incorporating additional pseudo-ground truth data from the internet, as shown in Table 6.5.

6.4.3 Pitch Statistics

Table 6.4 shows the pitch statistics from the broadcast videos. TCN with five TConv blocks is utilized to predict handedness and pitch position from the kinematic motion sequence of the pitcher. The ball release point is extracted using heuristics from the trajectory of the wrist position of the pitcher. The pitch velocity and release extension are then computed using mathematical functions that utilize ball release point and kinematic pose information.

Table 6.4a demonstrates that the TCN model achieves perfect accuracy (100%) in classifying the handedness, without misclassifications for right-handed or left-handed pitchers. The model demonstrates impressive performance in classifying pitch position, correctly identifying 95% of the stretch deliveries and 100% of windup deliveries as shown in Table 6.4b. The misclassification rate is low, with only 5% of stretch deliveries misclassified as windup, and no misclassification observed for windup deliveries.

The validity of the approach adapted to estimate the release point is examined and compared to alternative methods in Table 6.4c. The table shows that directly inferring the release point from a temporal network tends to perform poorly. A_x denotes the accuracy with x the number of frames as a margin in the table. Table 6.4d shows that our method estimates pitch velocity with superior performance compared to existing temporal networks. $A_{x\%}$ in Table 6.4d denotes the accuracy with $x\%$ as the margin. Finally, Table 6.4e presents the superior performance for estimating release extension. It can also be improved by studying the pitching stride length which is the distance covered between the spot where one foot hits the ground and the next time the same foot hits the ground again

[35]. Our method directly utilizes the release point and 3D pose information to calculate the extension, achieving accurate results compared to the baseline networks.

Figure 6.8: **Qualitative results.** Performance of the PitcherNet system in capturing various pitch statistics from the player tracklets. Here, *Pred.* denotes the prediction from the 3D pose information and GT denotes the ground truth game data.

The results provided in Figure 6.8 highlight the qualitative performance of the PitcherNet system in the MLBPitchDB dataset [15]. These visualizations underscore the effectiveness and robustness of our system in achieving accurate alignment with input pitch tracklets.

6.5 Summary

In this chapter, we introduce PitcherNet, an end-to-end deep learning system for kinematic-driven pitch analysis in baseball sports through robust 3D human modeling from broadcast videos. By overcoming challenges such as motion blur in low-resolution feeds, PitcherNet accurately identifies a range of pitch statistics, including pitch position, release point, pitch velocity, release extension, and pitcher handedness. This empowers players, coaches, and

fans to gain deeper insights into the technical nuances of pitching and unlock strategic advantages. Additionally, decoupling action from tracklets paves the way for reliable player identification, which holds significant potential for sports analytics and performance evaluation.

Chapter 7

Conclusion

This dissertation proposes a novel automated system that analyzes a pitcher's pose directly from broadcast videos. This system goes beyond simply identifying the pose; it extracts valuable pitch statistics derived from the pitcher's kinematic motion. A key innovation lies in the robustness of the underlying 3D human model. Unlike traditional methods, this model is designed to handle the challenges inherent in broadcast videos, including motion blur, occlusion, and low resolution. This enhanced robustness translates to reliable and accurate estimation of pitch statistics, enabling a comprehensive understanding of pitching mechanics in real-time.

The effectiveness of the proposed framework is validated through extensive experimentation. The D2A-HMR model demonstrates SOTA performance on various benchmark datasets, showcasing its ability to handle OOD data reliably. Furthermore, experiments confirm that PitcherNet's pitch analysis surpasses existing methods. Additionally, the focused augmentation strategy demonstrably improves the performance of existing human pose estimation models against motion blur effects. Finally, experiments validate the robustness of the role classification network used within PitcherNet for identifying pitchers within the broader video frame.

These combined achievements contribute significantly to the field of baseball analytics by enabling accurate and real-time analysis of pitching mechanics from broadcast video. The remainder of this chapter explores future research directions, the broader applicability of this work beyond baseball, and the overall impact of the research.

7.1 Potential for Future Research

The research presented in this dissertation provides a basis for future work in baseball sports analytics. Here are some exciting avenues for future work that can build upon the foundation laid by this dissertation:

1. **Advanced Biomechanical Analysis:** Investigation on how the data collected from the proposed system can identify potential overuse, injuries or develop personalized training programs to optimize mechanics and improve pitching efficiency.
2. **Multi-Camera Integration:** Incorporating data from multiple cameras to improve the accuracy and robustness of the 3D human model, especially when dealing with occlusions.
3. **Generalization to Other Sports:** Exploration of generalizing the proposed system for broader applicability in the sports domain, specifically in sports with similar throwing mechanics, such as cricket or javelin throw.
4. **Extended Pose Estimation:** Extending the human modeling technique to accurately include the fingers using techniques such as hand Model with Articulated and Non-rigid defOrmations (MANO) to predict the type of grip used for the pitch and the spin rate of the pitch.
5. **Enhanced Robustness Against Motion Blur:** Exploration of Generative Adversarial Networks (GANs) or blur compensation strategies or explicitly learning the motion representation to handle extreme blur conditions.

7.2 Applicability

The research presented in this thesis, focusing on 3D human modeling, pose estimation, and analysis in the context of baseball, has yielded a practical solution with broad applicability beyond the realm of sports. Our proposed technique demonstrates remarkable robustness in capturing accurate poses despite the low resolution, motion blur, and occlusion prevalent in broadcast videos. This paves the way for the direct application of this work to analyze the movements of athletes captured in broadcast videos across a range of sports.

Beyond the immediate benefits of these works as detailed in Section 1.2, this work has the potential to significantly impact other fields that require accurate information on

human movement. Specifically, in the healthcare sector, the proposed 3D human model can be utilized to analyze patient movement patterns (gait abnormalities, recovery progress) during physical therapy and rehabilitation. The application can further be extended to security and surveillance to detect suspicious activity based on gait patterns or specific gestures.

7.3 Impact

This dissertation presents several significant contributions that advance the field of 3D human modeling and its application to baseball analytics. A key contribution is the development of a novel framework called D2A-HMR, designed to estimate accurate 3D human models from broadcast video images. D2A-HMR addresses the challenge of OOD poses by incorporating both distribution awareness and depth information, leading to generalizable models that perform well even for poses not explicitly included in the training data.

Furthermore, this work introduces a focused augmentation strategy specifically designed to address the issue of motion blur, a prevalent challenge in broadcast videos. This strategy tackles a major hurdle in pose estimation by forcing the model to learn to handle blurry frames effectively.

To demonstrate the practical application of these advancements, this dissertation introduces PitcherNet– a novel system built upon D2A-HMR and the motion blur augmentation strategy. PitcherNet leverages the robust 3D human models generated by D2A-HMR to analyze baseball pitching mechanics directly from broadcast video footage. The system extracts accurate 3D pose information and utilizes it to calculate real-time pitch statistics. Additionally, PitcherNet proposes a role classification network, designed to decouple actions, enabling reliable pitcher identification within the broader broadcast video frame.

References

[1] Amit Agrawal, Yi Xu, and Ramesh Raskar. Invertible motion blur in video. In *ACM SIGGRAPH 2009 papers*, pages 1–8, 2009.

[2] D. Ahn, S. Kim, H. Hong, and B. Chul Ko. Star-transformer: A spatio-temporal cross attention transformer for human action recognition. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3319–3328, Los Alamitos, CA, USA, jan 2023. IEEE Computer Society.

[3] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021.

[4] Brett Allen, Brian Curless, and Zoran Popović. Articulated body deformation from range scan data. *ACM Transactions on Graphics (TOG)*, 21(3):612–619, 2002.

[5] Bavesh Balaji, Jerrin Bright, Harish Prakash, Yuhao Chen, David A. Clausi, and John Zelek. Jersey number recognition using keyframe identification from low-resolution broadcast videos. In *Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports*, MMSports '23, page 123–130, New York, NY, USA, 2023. Association for Computing Machinery.

[6] L. Ballan, M. Bertini, A. Del Bimbo, and W. Nunziati. Soccer players identification based on visual local features. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, page 258–265, New York, NY, USA, 2007. Association for Computing Machinery.

[7] Lamberto Ballan, Marco Bertini, A. Bimbo, and Walter Nunziati. Soccer players identification based on visual local features. In *ACM International Conference on Image and Video Retrieval*, 2007.

[8] Sermetcan Baysal and Pınar Duygulu. Sentioscope: A soccer player tracking system using model field particles. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(7):1350–1362, 2016.

[9] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[10] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.

[11] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[12] Benjamin Biggs, David Novotny, Sebastien Ehrhardt, Hanbyul Joo, Ben Graham, and Andrea Vedaldi. 3d multi-bodies: Fitting sets of plausible 3d human models to ambiguous image data. *Advances in Neural Information Processing Systems*, 33:20496–20507, 2020.

[13] Joel R Bock. Pitch sequence complexity and long-term pitcher performance. *Sports*, 3(1):40–55, 2015.

[14] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 561–578. Springer, 2016.

[15] Jerrin Bright, Bavesh Balaji, Harish Prakash, Yuhao Chen, David A Clausi, and John Zelek. Distribution and depth-aware transformers for 3d human mesh recovery. *arXiv preprint arXiv:2403.09063*, 2024.

[16] Jerrin Bright, Yuhao Chen, and John Zelek. Mitigating motion blur for robust 3d baseball player pose modeling for pitch analysis, 2023.

[17] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[18] Zixi Cai, Helmut Neher, Kanav Vats, David A Clausi, and John S. Zelek. Temporal hockey action recognition via pose and optical flows. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2543–2552, 2018.

[19] Z. Cao, G. Hidalgo 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*, 2019.

[20] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[21] Alvin Chan, Martin D. Levine, and Mehrsan Javan. Player identification in hockey broadcast videos. *Expert Systems with Applications*, 165:113891, 2021.

[22] Chien-Chang Chen, Chen Chang, Cheng-Shian Lin, Chien-Hua Chen, and I Cheng Chen. Video based basketball shooting prediction and pose suggestion system. *Multimedia Tools and Applications*, 82(18):27551–27570, 2023.

[23] Hua-Tsung Chen, Chien-Li Chou, Wen-Jiin Tsai, Suh-Yin Lee, and Jen-Yu Yu. Extraction and representation of human body for pitching style recognition in broadcast baseball video. In *2011 IEEE International Conference on Multimedia and Expo*, pages 1–4, 2011.

[24] Hanbyel Cho, Jaesung Ahn, Yooshin Cho, and Junmo Kim. Video inference for human mesh recovery with vision transformer. In *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–6, 2023.

[25] Hanbyel Cho, Yooshin Cho, Jaesung Ahn, and Junmo Kim. Implicit 3d human mesh recovery using consistency with pose and shape from unseen-view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21148–21158, 2023.

[26] Sunghyun Cho, Yasuyuki Matsushita, and Seungyong Lee. Removing non-uniform motion blur from images. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[27] Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1964–1973, 2021.

[28] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 769–787. Springer, 2020.

[29] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7024–7033, 2018.

[30] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[31] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[32] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *CoRR*, abs/1605.08803, 2016.

[33] Mehrnaz Fani, Helmut Neher, David A. Clausi, Alexander Wong, and John Zelek. Hockey action recognition via integrated stacked hourglass network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 85–93, 2017.

[34] Society for American Baseball Research. A guide to sabermetric research. *SABR*, 2021.

[35] Dave Fortenbaugh, Glenn Fleisig, and James Andrews. Baseball pitching biomechanics in relation to injury risk and performance. *Sports health*, 1:314–20, 07 2009.

[36] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[37] Sebastian Gerke, Karsten Müller, and Ralf Schäfer. Soccer jersey number recognition using convolutional neural networks. *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 734–741, 2015.

[38] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1711–1721, 2018.

[39] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. *arXiv preprint arXiv:2305.20091*, 2023.

[40] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7779–7788, 2020.

[41] Connor T. Heaton and Prasenjit Mitra. Learning to describe player form in the MLB. *CoRR*, abs/2109.05280, 2021.

[42] Kevin Hickey, Lina Zhou, and Jie Tao. Dissecting moneyball: Improving classification model interpretability in baseball pitch prediction. *Hawaii International Conference on System Sciences*, 2020.

[43] Geoffrey E Hinton. Neural networks for machine learning, lecture 6a: Overview of mini-batch gradient descent. *University of Toronto*, 2012.

[44] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014.

[45] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014.

[46] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9772–9781, 2021.

[47] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *bmvc*, volume 2, page 5. Aberystwyth, UK, 2010.

[48] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018.

[49] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5614–5623, 2019.

[50] JM Karnuta, BC Luu, HS Haeberle, PM Saluan, SJ Frangiamore, KL Stearns, and et al. Machine learning outperforms regression analysis to predict next-season major league baseball player injuries: epidemiology and validation of 13,982 player-years from performance and injury profile trends, 2000-2017. *Orthopaedic Journal of Sports Medicine*, 8, 2020.

[51] Matt Kelly. What is sabermetrics? modern analytics impact nearly every part of today's game. *MLB.com*, 2019.

[52] P. Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Why normalizing flows fail to detect out-of-distribution data. *ArXiv*, abs/2006.08545, 2020.

[53] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020.

[54] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11127–11137, 2021.

[55] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2252–2261, 2019.

[56] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4501–4510, 2019.

[57] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4501–4510, 2019.

[58] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4501–4510, 2019.

[59] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11605–11614, 2021.

[60] Kaan Koseler and Matthew Stephan. Machine learning applications in baseball: A systematic literature review. *Applied Artificial Intelligence*, 31:1–19, 02 2018.

[61] Dong Li, Ting Yao, Ling-Yu Duan, Tao Mei, and Yong Rui. Unified spatio-temporal attention networks for action recognition in videos. *IEEE Transactions on Multimedia*, 21(2):416–428, 2019.

[62] Gen Li, Shikun Xu, Xiang Liu, Lei Li, and Changhu Wang. Jersey number recognition with semi-supervised spatial transformer network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1864–18647, 2018.

[63] Jiahao Li, Zongxin Yang, Xiaohan Wang, Jianxin Ma, Chang Zhou, and Yi Yang. Jotr: 3d joint contrastive learning with transformers for occluded human mesh recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9110–9121, 2023.

[64] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11025–11034, 2021.

[65] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1944–1953, 2021.

[66] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019.

[67] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, and Pichao Wang. Lifting transformer for 3d human pose estimation in video. *CoRR*, abs/2103.14304, 2021.

[68] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13147–13156, 2022.

[69] Jing Lin, Ailing Zeng, Haoqian Wang, Lei Zhang, and Yu Li. One-stage 3d whole-body mesh recovery with component aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21159–21168, 2023.

[70] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1954–1963, 2021.

[71] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 238–247, 2022.

[72] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[73] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[74] Hengyue Liu and Bir Bhanu. Pose-guided r-cnn for jersey number recognition in sports. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2457–2466, 2019.

[75] Hengyue Liu and Bir Bhanu. Pose-guided r-cnn for jersey number recognition in sports. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2457–2466, 2019.

[76] Hongshan Liu, Colin Aderon, Noah Wagon, Huapu Liu, Steven MacCall, and Yu Gan. Deep learning-based automatic player identification and logging in american football videos. *arXiv preprint arXiv:2204.13809*, 2022.

[77] Junfa Liu, Juan Rojas, Zhijun Liang, Yihui Li, and Yisheng Guan. A graph attention spatio-temporal convolutional networks for 3d human pose estimation in video. *arXiv preprint arXiv:2003.14179*, 2020.

[78] Wu Liu, Chenggang Clarence Yan, Jiangyu Liu, and Huadong Ma. Deep learning based basketball video analysis for intelligent arena application. *Multimedia Tools and Applications*, 76:24983–25001, 2017.

[79] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. In *ACM Trans. Graph.*, volume 34, New York, NY, USA, 2015. Association for Computing Machinery.

[80] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[81] Jonathan Samuel Lumentut, Joshua Santoso, and In Kyu Park. Human motion deblurring using localized body prior. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[82] Haoyu Ma, Liangjian Chen, Deying Kong, Zhe Wang, Xingwei Liu, Hao Tang, Xiangyi Yan, Yusheng Xie, Shih-Yao Lin, and Xiaohui Xie. Transfusion: Cross-view fusion with transformer for 3d human pose estimation. *arXiv preprint arXiv:2110.09554*, 2021.

[83] Zahid Mahmood, Tauseef Ali, Shahid Khattak, Laiq Hasan, and Samee U. Khan. Automatic player detection and identification for sports entertainment applications. *Pattern Anal. Appl.*, 18(4):971–982, nov 2015.

[84] Joseph E Manzi, Brittany Dowling, Spencer Krichevsky, Nicholas LS Roberts, Suleiman Y Sudah, Jay Moran, Frank R Chen, Theodore Quan, Kyle W Morse,

and Joshua S Dines. Pitch-classifier model for professional pitchers utilizing 3d motion capture and machine learning algorithms. *Journal of Orthopaedics*, 49:140–147, 2024.

[85] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, Zhibin Wang, and Anton van den Hengel. Poseur: Direct human pose regression with transformers. In *European Conference on Computer Vision*, pages 72–88. Springer, 2022.

[86] MLB. Red sox vs. orioles game highlights (4/8/21) — mlb highlights. YouTube video, 2021.

[87] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 752–768. Springer, 2020.

[88] Tricia A Murray, Timothy D Cook, Sherry L Werner, Theodore F Schlegel, and Richard J Hawkins. The effects of extended play on professional baseball pitchers. *The American journal of sports medicine*, 29(2):137–142, 2001.

[89] Daiki Nasu and Makio Kashino. Impact of each release parameter on pitch location in baseball pitching. *Journal of sports sciences*, 39(10):1186–1191, 2021.

[90] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 483–499. Springer, 2016.

[91] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[92] Sakiko Oyama, Araceli Sosa, Rebekah Campbell, and Alexandra Correa. Reliability and validity of quantitative video analysis of baseball pitching motion. *Journal of applied biomechanics*, 33:64–68, 02 2017.

[93] Paschalis Panteleris and Antonis Argyros. Pe-former: Pose estimation transformer. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 3–14. Springer, 2022.

[94] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

[95] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019.

[96] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9054–9063, 2021.

[97] Zhongwei Qiu, Qiansheng Yang, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Chang Xu, Dongmei Fu, and Jingdong Wang. Psvt: End-to-end multi-person 3d pose and shape estimation with progressive video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21254–21263, 2023.

[98] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021.

[99] Xiaofeng Ren, Alexander C Berg, and Jitendra Malik. Recovering human body configurations using pairwise constraints between parts. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 824–831. IEEE, 2005.

[100] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. *CoRR*, abs/2108.06428, 2021.

[101] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[102] Joshua Santoso, In Kyu Park, et al. Holistic 3d body reconstruction from a blurred single image. *IEEE Access*, 10:115399–115410, 2022.

[103] Donna Moxley Scarborough, Nicholas K Leonard, Lucas W Mayer, Luke S Oh, and Eric M Berkson. The association of baseball pitch delivery and kinematic sequence on stresses at the shoulder and elbow joints. *Journal of Sports Science & Medicine*, 20(1):94, 2021.

[104] Arda Senocak, Tae-Hyun Oh, Junsik Kim, and In So Kweon. Part-based player identification using deep convolutional representation and multi-scale pooling. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1813–18137, 2018.

[105] Zehong Shen, Zhi Cen, Sida Peng, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Learning human mesh recovery in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17038–17047, 2023.

[106] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[107] Glenn Sidle and Hien Tran. Using multi-class classification methods to predict baseball pitch types. *J. Sports Anal.*, 4(1):85–93, February 2018.

[108] Glenn Sidle and Hien Tran. Using multi-class classification methods to predict baseball pitch types. *Journal of Sports Analytics*, 4(1):85–93, 2018.

[109] T Sieberth, R Wackrow, and JH Chandler. Motion blur disturbs–the influence of motion-blurred images in photogrammetry. *The Photogrammetric Record*, 29(148):434–453, 2014.

[110] Xuhui Song and Linyuan Fan. Human posture recognition and estimation method based on 3d multiview basketball sports dataset. *Complexity*, 2021:1–10, 2021.

[111] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[112] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. Monocular, One-stage, Regression of Multiple 3D People. In *ICCV*, 2021.

[113] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[114] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[115] Kanav Vats, William J. McNally, Pascale Walters, David A Clausi, and John S. Zelek. Ice hockey player identification via transformers and weakly supervised learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3450–3459, 2021.

[116] Kanav Vats, Pascale Walters, Mehrnaz Fani, David A. Clausi, and John S. Zelek. Player tracking and identification in ice hockey. *Expert Systems with Applications*, 213:119250, 2023.

[117] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2598–2606, 2018.

[118] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European conference on computer vision (ECCV)*, pages 601–617, 2018.

[119] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018.

[120] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016.

[121] David Whiteside, Douglas N Martini, Ronald F Zernicke, and Grant C Goulet. Ball speed and release consistency predict pitching success in major league baseball. *The Journal of Strength & Conditioning Research*, 30(7):1787–1795, 2016.

[122] N. Woodward. A decision tree approach to pitch prediction. *The Hardball Times*, 2014.

[123] Z. Xing, Q. Dai, H. Hu, J. Chen, Z. Wu, and Y. Jiang. Svformer: Semi-supervised video transformer for action recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18816–18826, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society.

[124] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J Black. Econ: Explicit clothed humans optimized via normal integration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 512–523, 2023.

[125] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems*, 35:38571–38584, 2022.

[126] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024.

[127] A. Yao, J. Gall, and L. Van Gool. Coupled action recognition and pose estimation from multiple views. *Int J Comput Vis*, 100:16–37, 2012.

[128] Angela Yao, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Does human action recognition benefit from pose estimation?". In *Proceedings of the 22nd British machine vision conference-BMVC 2011*. BMV press, 2011.

[129] Hongwei Yi, Chun-Hao P Huang, Shashank Tripathi, Lea Hering, Justus Thies, and Michael J Black. Mime: Human-aware 3d scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12965–12976, 2023.

[130] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly supervised 3d human pose and shape reconstruction with normalizing flows. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 465–481. Springer, 2020.

[131] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11446–11456, 2021.

[132] Ruiheng Zhang, Lingxiang Wu, Yukun Yang, Wanneng Wu, Yueqiang Chen, and Min Xu. Multi-camera multi-player tracking with deep player identification in sports video. *Pattern Recognition*, 102:107260, 2020.

[133] Yiming Zhao, Denys Rozumnyi, Jie Song, Otmar Hilliges, Marc Pollefeys, and Martin R Oswald. Human from blur: Human pose tracking from blurry images. *arXiv preprint arXiv:2303.17209*, 2023.

[134] Nikolaos Zioulis and James F O'Brien. Kbody: Towards general, robust, and aligned monocular whole-body estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6214–6224, 2023.