Human pose estimation in 3D for working environment

Yue Zhu

École des Ponts ParisTech

THÈSE DE DOCTORAT

de l'École des Ponts ParisTech

Human pose estimation in 3D for working environment

École doctorale N°532, Mathemetiques et Sciences et Technologies de l'Information et de la Communication (MSTIC)

Informatique

Laboratoire d'Informatique Gaspard-Monge (LIGM), UMR 8049, IMAGINE

Thèse soutenue le 19 april 2024, par

Yue Zhu

Composition du jury:

Hilde Kuehne Professeure, Université de Bonn	*Examinatrice*
Hedi Tabia Professeur, Université d'Evry	*Examinateur*
Catherine Achard Professeure, Université Sorbonne	*Rapportrice*
Nicolas Thome Professeur, Université Sorbonne	*Rapporteur*
David Picard Professeur, Ecole des Ponts ParisTech	*Directeur de thèse*
Louis Meurisse Ingénieur, Ergonova Conseil	*Co-encadrant de thèse*

Dedication

This thesis is dedicated to all people that help me throughout my Ph.D education.

Thank you all for your accompany.

Acknowledgements

First printing, February 2024

Résumé

Les objectifs de la thèse sont de développer des méthodes et des cadres d'analyse de la posture humaine 3D en environnement de travail pour l'ergonomie.

L'ergonomie est une discipline qui consiste à comprendre le fonctionnement du corps lorsqu'il travaille pour l'objectif de préserver la santé des opérateurs tout en permettant l'atteinte de la qualité attendue. Les postures des opérateurs au poste de travail sont un des facteurs d'apparition des maladies professionnelles, et la caractérisation d'une posture est une étape du pré-diagnostic d'une situation de travail. Les méthodes d'intelligence artificielle autour de l'estimation 3D de la pose humaine pour détecter les postures inadaptées au travail peuvent ainsi aider l'ergonome à établir son diagnostic sur un grand nombre de données.

Cette thèse propose trois travaux autour de l'estimation de pose humaine 3D pour s'attaquer aux difficultés de mise en œuvre en environnement non contraint tels que les postes de travail.

La première contribution propose un algorithme synthétique de génération de poses humaines en 3D. Nous abordons le problème de l'écart de domaine selon lequel les scénarios de travail a plus de variété d'actions et d'environment que les données de recherche publique. Ce travail présente un algorithme qui permet de générer des squelettes humains 3D synthétiques pendant l'entraînement de réseau des neurons, suivant une distribution de type arbre de Markov qui évolue au fil du temps pour créer des nouvelles postures. Ce travail propose également un processus d'entraînement multi-vues sans échelle basé sur des données purement synthétiques générées à partir de quelques postures initiales. Nous évaluons notre approche sur les deux ensembles de données de référence et obtenons des résultats prometteurs dans une configuration sans aucune donnée réelle.

Le deuxième travail propose un cadre de création d'annotations 3D du corps entier à partir d'images multi-vues ainsi qu'un benchmark construit sur la base de ce cadre. Les données couramment utilisées ne comportent normalement qu'une vingtaine d'articulations, ce qui n'est pas suffisant pour qu'un ergonome puisse mesurer certains aspects comme les angles de supination-pronation, c'est pourquoi nouw pro-

posons un squelette du corps entier compte 133 articulations, capables de contenir les informations nécessaires. Le cadre de création d'annotations contient 3 étapes allant de la reconstruction géométrique 3D multi-vues à la completion des squelettes incomplets, et enfin au raffinement main/visage par diffusion. Avec ce cadre, nous introduisons 3 ensembles de données en tant qu'extensions des ensembles de données Human3.6M, CMU-Panoptic et MPI-INF-3DHP existants avec des annotations de points clés 2D et 3D du corps entier pour le corps, le visage et les mains. Un benchmark de trois tâches est proposé sur la base de l'extension du corps entier de Human3.6M.

Le troisième travail propose un algorithme qui permet une prediction continue des poses humaines à travers le temps avec des images d'entrée très limitées pour aborder des séquences vidéo potentiellement corrompues dans un environnement sans contrainte où les travailleurs ne sont pas toujours observés ou même à l'écran en raison de leurs mouvements. Ce travail propose une nouvelle approche qui modélise le mouvement humain comme une fonction continue mise en œuvre par un réseau neuronal, semblable à des représentations neuronales implicites. Nous effectuons une comparaison complète de cette approche avec des méthodes de prédiction de mouvement de pointe sur trois ensembles de données populaires, démontrant des améliorations significatives par rapport aux lignes de base dans la plupart des cas.

Enfin, nous avons réalisé un démonstrateur qui effectue une estimation de la pose humaine en 2D et 3D, ainsi qu'une détection des poses critiques pour une analyse ergonomique, capable d'analyse rapide même sur un ordinateur équipé uniquement d'un CPU.

Abstract

The objectives of the thesis is to develop the methods and frameworks to analysis 3D human postures in the working environment for ergonomic propose.

Ergonomics is a discipline which consists of understanding body work with the objective of preserving the health of operators while allowing achievement of the expected quality. The postures of operators at work stations are one of the factors in the appearance of occupational diseases, and the characterization of a posture is a step in the pre-diagnosis of a work situation. Artificial intelligence methods using 3D human pose estimation to detect unsuitable postures at work could help ergonomist to establish their diagnosis on a large quantity of data.

This thesis proposes three works on 3D human pose estimation to attack the difficulties of unconstrained environment such as work stations.

The first work proposes a synthetic 3D human pose generation algorithm for training 2D to 3D human pose lifting. We tackle with the domain gap problem between public research data which have constrained environment plus limited number of action and unconstrained working environment data with much more variety of actions. This work presents an algorithm which allows to generate synthetic 3D human skeletons on the fly during the training, following a Markov-tree type distribution which evolve through out time to create unseen poses. This work also proposes a scaleless multi-view training process based on purely synthetic data generated from a few initial poses. We evaluate our approach on two benchmark datasets and achieve promising results in a zero shot setup.

The second work proposes a framework of making 3D wholebody annotations from multi-view image data as well as a few datasets and a benchmark built based on this framework to tackle with the skeleton model capability problem. Public research data normally only has around 20 keypoints which is not enough for ergonomist to measure certain aspects like supination-pronation angles, while the wholebody skeleton has 133 keypoints, capable for obtaining the necessary information. The annotation-making framework contains 3 steps from multi-view 3D geometry reconstruction, to incomplete skeleton completion, and finally hand/face refinement

through diffusion. With this framework, we introduce 3 datasets as extensions of existing Human3.6M, CMU-Panoptic and MPI-INF-3DHP datasets with wholebody 2D and 3D keypoint annotations for body, face, and hands. A benchmark of three tasks is proposed based on Human3.6M wholebody extension: 3D whole-body lifting from complete 2D keypoints, from incomplete 2D keypoints, and from monocular images.

The third work proposes an algorithm which allows a continuous estimation of human poses through time with very limited input frames to tackle potential corrupted video sequences in unconstrained environment where the workers are not always observed or even in the screen due to their movements. This work proposes a novel approach that models human motion as a continuous function implemented by a neural network, akin to neural implicit representations. We conduct a comprehensive comparison of this approach with state-of-the-art motion prediction methods on three popular datasets, demonstrating significant improvements over the baselines in most cases.

Finally, we made a demonstration that perform 2D and 3D human pose estimation, as well as critical pose detection for ergonomic analysis, capable of fast processing even on a CPU-only computer.

Résumé étendu

Introduction

Les objectifs de la thèse sont de développer des méthodes et des cadres d'analyse de la posture humaine 3D en environnement de travail pour l'ergonomie.

L'ergonomie est une discipline qui consiste à comprendre le fonctionnement du corps lorsqu'il travaille pour l'objectif de préserver la santé des opérateurs tout en permettant l'atteinte de la qualité attendue. Ceci est plus critique pour les personnes qui effectuent un travail physique pénible. Une façon de détecter de tels dommages potentiels sur leur corps consiste à suivre la posture de l'opérateur, ce qui permet une analyse plus profondre de son corps et de ses actions. Le développement rapide de l'intelligence artificielle et les progrès significatifs de la vision par ordinateur ont permis la réalisation d'une détection et d'une analyse automatiques dans un environnement complexe. Un tel algorithme de détection et d'analyse est appelé **Estimation de la Posture Humaine**. Cependant, même si l'estimation de la posture humaine est un problème bien connu et largement développé en vision par ordinateur, il reste un problème complexe dont les solutions existantes ne répondent pas à tous les besoins de cette thèse.

Système choisi pour le problème des environnements contrôlés et naturels

Notre thèse nécessite la capacité d'être utilisé dans l'environnement de travail, qui contient à la fois des scénarios intérieurs et extérieurs, que l'on peut appeler 'natural' comme dans la vraie vie, alors que la plupart des méthodes plus récentes sur l'estimation de la posture humaine en 3D sont basées sur des scénarios intérieurs capturées dans un environnement contrôlé avec les arrière-plans limité et les types de postures prédéfini en raison d'une limitation matérielle. Afin d'obtenir des positions précises de posture humaine en 3D, un processus spécifique appelé 'capture de mouvement' (Motion Capture) est appliqué pour enregistrer les mouvements humains. Grâce à un ensemble de capteurs installés sur le corps des acteurs, les positions 3D

de ces capteurs peuvent être enregistrées par l'ordinateur. Une telle mesure peut être à la fois précise et en temps réel, mais le processus de capture de mouvement nécessite un espace, un matériel et un logiciel spécifiques pour permettre le processus, ce qui en fait un outil adapté uniquement dans un laboratoire, est ce n'est pas approprié pour un système compatible avec les scénarios extérieurs dont nous avons besoin.

Pour résoudre ce problème, l'ensemble du système se décompose en 3 étapes consécutives qui s'appliquent les unes après les autres.(Voir Figure 1).

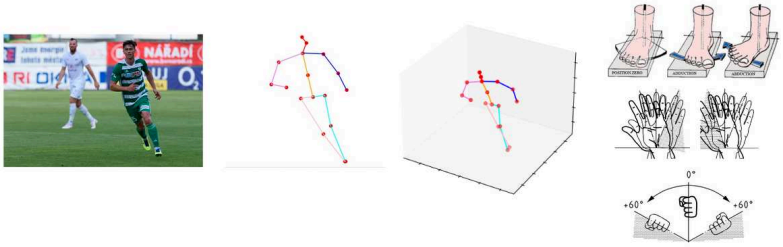

Figure 1: Un exemple de l'apparence des résultats entre chaque étape. A partir d'une image, l'estimation de posture 2D, l'augmentation de posture 2D à 3D et le calcul géométrique fourniront respectivement les résultats affichés dans les images successives. Source des images: **football**: [Kreiss et al., 2021] OpenPifPaf:Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association, **angles pied/main**: Template squelette 3D_v2_13012021.pptx de Ergonova Conseil.

1. Premièrement, le système utilise une seule image 2D capturée par la caméra comme entrée et renvoie les coordonnées 2D des articulations de chaque humain dans l'image. Cette étape correspond à **l'estimation de la posture humaine 2D à partir d'une image**.

2. Ensuite, pour chaque humain dans l'image, le système prend en entrée les coordonnées 2D des articulations et renvoie leurs coordonnées 3D dans l'espace de la caméra. Cette étape correspond à **l'estimation de la posture humaine 3D à partir de la posture humaine 2D**.

3. Enfin, pour chaque ensemble de coordonnées 3D correspondant à un même individu, le système calcule les angles. Cette étape est appelée **calcul géométrique**,

La raison pour laquelle nous effectuons d'abord une estimation de la posture humaine en 2D, puis augmenter la posture humaine 2D en 3D est due au fait qu'enregistrer des coordonnées précises de la posture humaine en 2D est beaucoup plus simple,

aussi simple que de cliquer manuellement sur l'image pour indiquer quel pixel correspond à un articulation du corps. En effet, il existe déjà plusieurs ensembles de données extérieurs 'naturels' de postures humaines 2D avec annotations de coordonnées. En revanche, augmenter la posture humaine 2D en 3D ne nécessite que les calculs entre les coordonnées 2D et les coordonnées 3D, ce qui signifie que l'environnement d'arrière-plan n'affecte pas cette étape. Donc, en divisant l'estimation de la posture humaine 3D en deux étapes, nous réussissons à éviter la limite de l'environnement contrôlés causé par des ensembles de données 3D existants.

Première contribution pour le problème des actions simples et professionnels

A part de la grande diversité des environnements dans différentes conditions de travail, il existe également une grande diversité d'actions différentes que les individus effectuent au cours du travail. De la posture générale debout, assise ou allongée au sol, aux actions des mains allant du levage, de l'opération devant la tête ou de l'affaissement naturel, ces actions peuvent être diverses selon le type de travail. Malheureusement, la plupart des ensembles de données existants sont basés sur les actions simples les plus fréquentes, et aucun de ces ensembles de données ne couvre pas les postures humaines de différents emplois professionnels. Nous avons encore besoin d'une forte généralisation de notre système pour couvrir les différentes actions.

Pour résoudre ce problème, nous proposons notre première contribution, qui propose un algorithme synthétique de génération de postures humaines en 3D et l'entraînement des réseaux de neurones. (Voir Figure 2)

L'algorithme de génération contient les idées suivantes. Selon la nature d'un corps humain, nous supposons que la position d'une articulation (appelée enfant) dépend de la position de l'articulation qui lui est directement connectée mais plus proche du corps central au sens géodésique (appelée 'parent'). Nous définissons donc l'articulation du bassin comme articulation de base et une structure d'arbre est appliquée pour générer les articulations une par une. Nous générons l'articulation enfant dans un système de coordonnées sphériques local centré sur son articula-

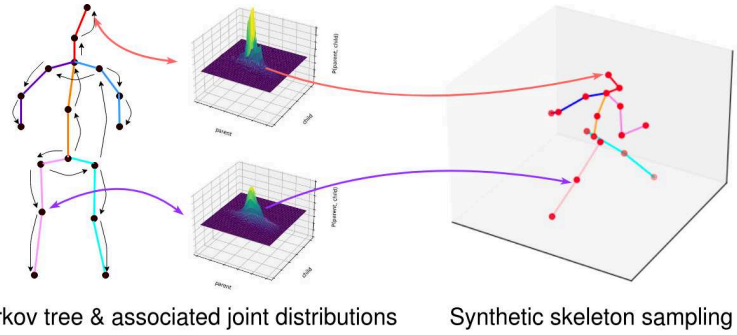

Figure 2: L'idée générale de notre méthode de génération synthétique: utiliser un arbre probabiliste hiérarchique et sa distribution par joint pour générer des postures humaines synthétiques 3D réalistes.

tion parent car chaque branche du corps humain a une longueur fixe quel que soit le mouvement, donc la nouvelle position de l'articulation enfant peut être paramétrée avec l'angle polaire et angle azimutal. Notre système de coordonnées sphériques local est également entièrement bijectif avec le système de coordonnées cartésiennes global, permettant la transformation simple entre l'espace de coordonnées sphériques de génération et l'espace de coordonnées cartésiennes commun.

Désormais, générer une posture humaine dans notre système de coordonnées sphériques local équivaut à générer un ensemble d'angles. Nous proposons de choisir ces valeurs à partir d'une distribution qui se rapproche de celle de postures humaines réelles. Nous limitons l'interval d'angles pour chaque articulation en fonction de ce qui est en moyenne biologiquement réalisable. Puisque la probabilité que chaque angle de l'enfant ne puisse pas rester la même lorsque le parent la valeur commune change, car ce dernier indiquant normalement une action différente, nous proposons de choisir les valeurs d'angle par rapport à une distribution conditionnelle $P(X_{enfant}|X_{parent})$. Cela produit une arbre de Markov pour les angles.

L'étape suivante consiste à estimer une distribution qui peut se rapprocher de la distribution réelle des postures humaines 3D et à partir de laquelle notre modèle peut choisir les angles. Sous la contrainte de ne touche pas des données réelles 3D et purement 3D synthétiques, nous avons choisi d'utiliser un nombre limité de postures réelles 2D et en les augmenter 'manuellement' en 3D pour estimer notre distribution. Nous choisissons une procédure en 3 étapes pour obtenir notre distribution. Nous choisissons 10 postures 2D avec une grande variance comme une base. Nous les

augmentons semi-automatique de 2D en 3D, et créons la distribution initiale à partir de ces postures. Nous diffusons les probabilités à l'intérieur du graphe de distribution comme la diffusion de chaleur pour augmenter la variété de génération.

Le modèle d'entraînement du réseau est dans la Figure 3. Nous évaluons notre approche sur les deux ensembles de données de référence et obtenons des résultats prometteurs dans une configuration sans aucune donnée réelle.

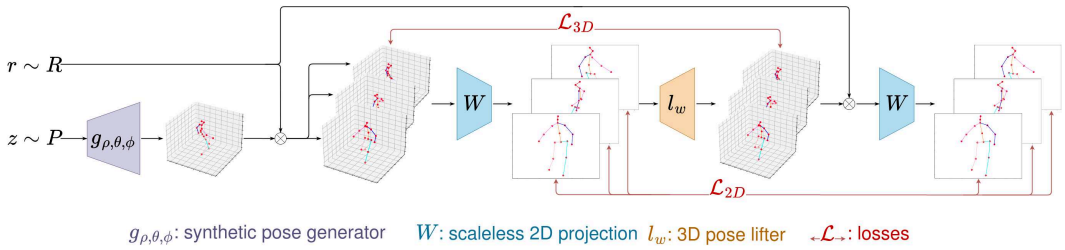

Figure 3: Notre processus d'entraînement avec des données synthétiques. Notre générateur g génère une pose humaine 3D suivant les distributions données P. Il est appliqué avec plusieurs r générés aléatoirement pour projeter dans différentes vues des caméras. Le projecteur W les projet en coordonnées 2D et ce sont les entrées du réseau. Les poses 3D estimées en sortie sont appliquées avec une loss de supervision 3D $\mathscr{L}_{3D}$, ainsi qu'une loss de reprojection 2D à vue croisée $\mathscr{L}_{2D}$.

Deuxième contribution pour le problème de la capacité du modèle squelette

La capacité du modèle squelette représente la quantité d'informations que nous pouvons dériver du squelette que nous choisissons, qui est liée au calcul géométrique dans l'étape suivante. Cependant, les méthodes courantes de l'état de l'art présentent deux inconvénients. Premièrement, ils n'utilisent qu'une vingtaine d'articulations qui représentent uniquement les articulations les plus critiques du corps humain. Bien que ces articulations soient capables de calculer l'angle comme la flexion et l'extension, elles ne sont pas capables de calculer l'angle de supination-pronation, ce qui rend ces conceptions de squelette moins préférables dans cette thèse. Le deuxième inconvénient est que les méthodes les plus récentes traitent le visage et les mains indépendamment. Ils apprennent à reconnaître séparément les articulations du corps, des mains et du visage, puis à les combiner en une seule humaine, mais nous préférons une structure combinée de toutes les informations pour faciliter le

calcul. Nous devons donc trouver une disposition de squelette adaptée à notre projet.

Pour résoudre ce problème, nous proposons notre deuxième contribution, qui propose un cadre de création d'annotations 3D du corps entier avec 133 articulation à partir d'images multi-vues ainsi qu'un benchmark construit sur la base de ce cadre. Le modèle squelette que nous choisissons et qui combine toutes les parties du corps en un existe déjà en 2D (mais pas en 3D avant notre travail), appelé COCO-Wholebody[1], qui fournit une disposition de 133 articulations, composé de 17 articulations du corps, 6 articulations des pieds, 68 articulations du visage et 42 articulations des mains. Et le travaux existant, Openpifpaf[2], disposent déjà d'un modèle et de poids bien pré-entraînés pour la détection de pose du corps entier en 2D, ce qui facilite le début de notre travail.

[1] [Jin et al., 2020]

Whole-Body Human Pose Estimation in the Wild

[2] [Kreiss et al., 2021]

OpenPifPaf:Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association

Nous exécutons un détection des corps entier en 2D par OpenPifPaf sur les 4 vues de caméra différentes à partir d'images multivues. Comme les caméras sont bien calibrées, nous pouvons reconstruire les articulation en 3D à l'aide d'un algorithme de géométrie multi-vues. Malheureusement, le détecteur OpenPifPaf ne predit pas des articulations en raison d'occlusions (mains, pieds) ou de points de vue défavorables de la caméra (face vers l'arrière). Cependant, la configuration à 4 vues nous permet de récupérer certains articulation manquants dans un vue mais visible dans les autre, et d'obtenir une pose complète du corps entier en 3D, à condition que chaque articulation apparaisse dans au moins deux vues non opposées. En utilisant cette méthode, nous avons obtenu 11,426 postures 3D complètes du corps entier avec les 133 articuations et 26,333 postures 3D incomplètes du corps entier où tous les points clés apparaissent dans au moins une vue, ce qui donne un total de 37,759 postures 3D du corps entier avec chaque articulation au moins vérifiable en 2D.

Afin de compléter les 26,333 postures incomplètes du corps entier en 3D, nous développons un réseau de complétion comme dans la Figure 4. Nous utilisont l'architecture Transformer[3] car ils peuvent facilement gérer les dépendances conditionnelles introduites par la topologie du squelette via le masquage. Puisque chaque squelette comporte toujours exactement 133 articulation, qui peuvent être considérés comme 133 jetons avec 3 valeurs. Les valeurs des jetons sont étendues de 3 coordonnées à $3 \times 16 = 48$ à l'aide du codage de Fourier. Nous utilisons un codage positionnel ap-

[3] [Vaswani et al., 2017]

Attention is all you need

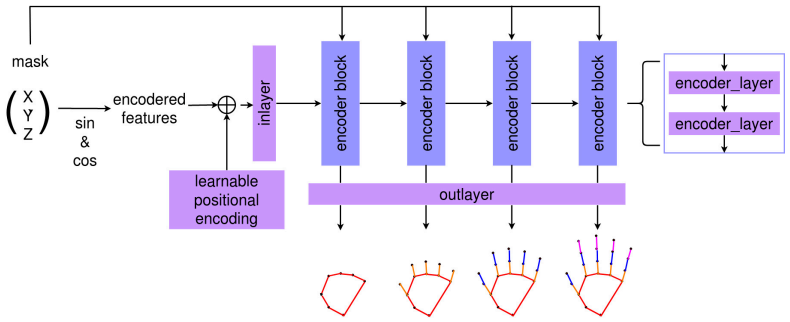

Figure 4: Le réseau de complétion se compose d'une couche d'entrée linéaire, de 4 blocs d'encodeurs de transformateur (chacun d'eux contenant 2 couches d'encodeurs de transformateur) et d'une couche de sortie linéaire. À la fin de chaque bloc d'encodeur, les valeurs sont décodées par la couche de sortie dans une stratégie curriculaire.

prenable puisque chaque articuation est identifié de manière unique. Nous entraînons le réseau de complétion sur les 11,426 squelettes complets en utilisant une stratégie d'encodeur automatique masqué [4] où les articulations manquants sont masqués en entrée et seront prédits à l'aide des points clés non masqués. La stratégie de masquage est la suivante: avec 50% de chances, nous effectuons un masque par articulations où chaqu'un a 15% de chance d'être masqué, et avec les 50% de chances restantes, nous effectuons un masque par blocs dans lequel soit le corps, la main gauche, la main droite, la partie gauche ou droite du visage sont masqués avec une probabilité uniforme. Pour faciliter le processus d'apprentissage, nous introduisons une approche curriculaire. Nous calculons la loss à différents niveaux en suivant une hiérarchie où les premiers niveaux considèrent uniquement les articulations plus proches de la corps, tandis que les niveaux ultérieurs considèrent les articulations plus déformables qui dépendent fortement de leurs parents. Les résultats du réseau de complétion sur les parties du corps manquantes sont visuellement réalistes. Cependant, comme le réseau de complétion ne s'appuie pas sur le contenu de l'image, sa sortie ne s'aligne pas toujours sur l'image et peut refléter uniquement les poses les plus courantes.

Pour corriger le problème d'alignement, nous proposons un autre réseau de neurones qui raffine la position 2D des articulation sur le visage et les mains. Nous nous appuyons sur des modèles de diffusion conditionnelle [5] récents. Pendant l'entraînement, nous ajoutons du bruit gaussien aux postures de vérité terraine avec une variance croissante de 5 à 25 pixels, et les annotons comme étape $t = 1...5$ (l'étape $t = 0$ est la

[4] [He et al., 2022b]

Masked autoencoders are scalable vision learners

[5] [Ho et al., 2020]

Denoising diffusion probabilistic models

vérité terraine). Le réseau apprend à prédire la pose à l'étape t étant donné l'image et l'étape la plus bruyante $t+1$ avec une loss de supervision 2D (Voir Figure 5). Nous exécutons les réseaux de raffinement entraînés sur les projections 2D des postures 3D prédites par notre réseau de complétion. Pour chaque squelette 3D, nous le projetons dans les 4 vues différentes. Nous recadrons ensuite les régions autour des mains et du visage et débruitons les prédictions correspondantes en utilisant le réseau de raffinement avec 10 itérations pour obtenir postures 2D raffinées dans chacune des 4 vues. On pratique une autre reconstruction géométrique pour soulever en 3D. Grâce à cette méthode, on obtient 151,036 triplets de points clés 3D du corps entier, image correspondante et points clés projetés en 2D à partir de l'ensemble d'origine. (Voir Figure 6)

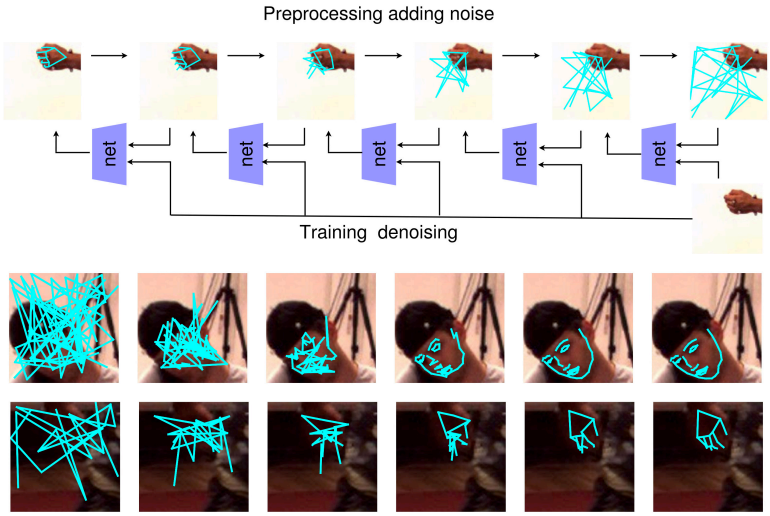

Figure 5: Réseau de raffinement et du processus d'entraînement. Un bruit gaussien est ajouté aux coordonnées de vérité terrain avec une variance croissante, et le réseau est entraîné de manière itérative pour récupérer les coordonnées les moins bruyantes. Les inférences convergent presque vers les emplacements corrects en 5 itérations.

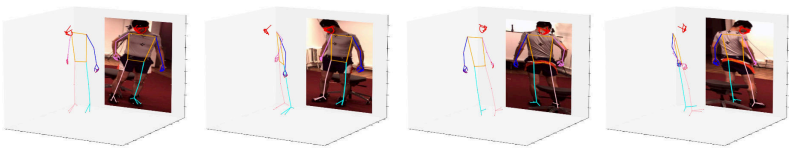

Figure 6: L'ensemble de nos données contient 133 annotations des articulations du corps entier en 3D ainsi que leurs projections respectives en 2D.

Avec ce cadre, nous introduisons 3 ensembles de données en tant qu'extensions des ensembles de données Human3.6M, CMU-Panoptic et MPI-INF-3DHP existants

avec des annotations des articulations 2D et 3D du corps entier pour le corps, le visage et les mains. Un benchmark de 3 tâches est proposé sur la base de l'extension du corps entier de Human3.6M, concernant estimation des postures du corps entier 3D depuis corps entier 2D complet, depuis coprs entier 2D incomplet et depuis un image. Plusieurs états des arts sont testées sur ces 3 tâches et il est démontré qu'il reste encore place à l'amélioration sur ce modèle de squelette plus difficile.

Troisième contribution pour le problème d'occlusion à l'intérieur d'une séquence de mouvement

Des travaux antérieurs nous montrent que les occlusions peuvent grandement affecter l'estimation de la posture humaine, alors qu'elles se produisent très souvent dans un environnement de travail complexe. Nous voulons étudier le cas où le mouvement humain à l'intérieur des séquence de mouvement est complètement occulté et inobservé, et de les récupérer à l'aide de début et de fin où l'humain est encore visible.

Pour résoudre ce problème, nous proposons notre troisième contribution, qui propose une nouvelle approche qui modélise le mouvement humain comme une fonction continue mise en œuvre par un réseau neuronal, qui permet une prediction continue des postures humaines à travers le temps. Nous formulons deux réseaux, $\mathscr{F}$ et $\mathscr{G}$, comme illustré dans Figure 7. Un réseau d'encodeurs $\mathscr{F}$ prend les M frames observées en entrée et les code dans un vecteur de caractéristiques latentes z. Le réseau $\mathscr{G}$ prend le vecteur z conditionné au temps t comme entrée et génère la pose correspondante $\hat{X}_t$ au temps t. Dans notre modèle, une fois les frames d'entrée sont codées en z, $\mathscr{G}$ devient fonction de l'unique variable t, permettant l'expression de l'ensemble du mouvement en faisant uniquement varier le temps.

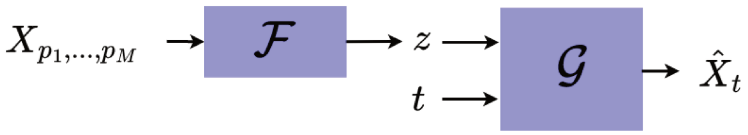

Figure 7: Notre modèle d'interpolation de mouvement

Nous effectuons une comparaison complète de cette approche avec des méthodes de prédiction de mouvement de pointe sur trois ensembles de données populaires, démontrant des améliorations significatives par rapport aux lignes de base dans la plupart des cas.

Un prototype qui combine tout

Enfin, nous avons réalisé un prototype qui effectue une estimation de la posture humaine en corps entier 2D et 3D, ainsi qu'une détection des postures critiques pour une analyse ergonomique, capable d'analyse rapide même sur un ordinateur équipé uniquement d'un CPU.

Nous utilisons un réseau entraîné pour estimer le corps entier 2D à partir d'une image, puis un autre réseau entraîné pour augmenter un corps entier 2D incomplet vers un corps entier 3D complét, et ils calculent les angles et les comparent aux zones de sécurité introduites par les ergonomes. L'interface ressemble à Figure 8 et l'ensemble de l'algorithme fonctionne sur un ordinateur sans GPU à environ 1,3 image-par-second.

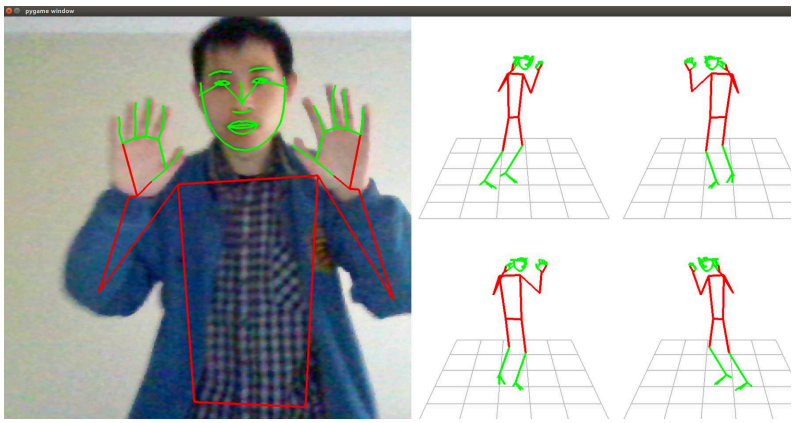

Figure 8: Un exemple d'écran de démonstration capturant l'auteur de cette thèse et restituant le squelette 2D et 3D sur l'interface. On voit que même la partie inférieure du corps n'est pas dans une image 2D ainsi que le squelette 2D, l'algorithme parvient quand même à prédire le squelette 3D complet à droite. Les corps sont rendus en rouge parce que l'auteur penche le corps vers l'avant, et maintenir cette pose n'est pas bon pour la colonne vertébrale, un problème existe pour de nombreuses personnes qui sont toujours assises et travaillent devant l'ordinateur.

Publications

This thesis draws heavily on earlier work and writing in the following author's papers:

Decanus to Legatus: Synthetic training for 2D-3D human pose lifting. (Yue Zhu, David Picard), Proceedings of the Asian Conference on Computer Vision (ACCV), 2022, pp. 2848-2865

H3WB: Human3.6M 3D WholeBody Dataset and Benchmark. (Yue Zhu, Nermin Samet, David Picard), Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 20166-20177

PIUS: Pose Interpolation at extremely low and Uneven framerateS. (Yue Zhu, Nermin Samet, David Picard), In submission, 2023.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Figure 1.1: Many works that require people to maintain the same posture for a long time can be a harmful factor to the body after a long period of time. Image source: [Mejean, 2020] Les operateurs n'ont peut-etre pas raison mais ils ont leurs raisons.

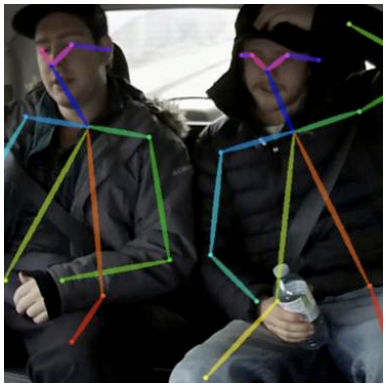

Figure 1.2: "Pose estimation is a computer vision task where the goal is to detect the position and orientation of a person or an object. Usually, this is done by predicting the location of specific keypoints" Image source: [paperswithcode, 2020] Pose estimation

Ergometry is a discipline that studies the measurement of the physical work of the body, in order to make such work more sustainable and keep the body in good health. This is more critical for people who perform heavy physical labor, as they often suffer from bodily illnesses due to poor working conditions and unscientific behaviour during work. (See Figure 1.1) One way to detect such potential damage to their body is to track the operator's posture, which allows for a more in-depth analysis of their body and actions. In this context, the characteristic of a human posture is defined as the relative positions of each body part in space at one moment during the work, which can be simply captured in an image with a camera, or even more complex, with specific motion capture equipment with sensors installed on the operator's body. Although the latter method is not viable for ordinary people because it requires special equipment and environment, the former is more likely to be applied in everyday life. The critical data includes body angles, arm and leg supination and pronation, actions over time, and more. On the other hand, the fast development of artificial intelligence and recent significant progress in computer vision have enabled the realization of automatic detection and analysis in a complex environment. One of the biggest examples of such a system is autonomous car driving, in which AI detects and decides the direction and speed of the car. As for ergometry, such a detection and analysis algorithm is called **Human Pose Estimation** (See Figure 1.2).

Human pose estimation is a topic that has been deeply studied in computer vision, mainly thanks to the rapid development of deep neural networks. The use of a human pose estimation algorithm to facilitate ergometric study is very promising, as it allows fast and systematic analysis of a huge amount of existing data. In fact, such automatic analysis of human poses in the working environment over a significant period of time (e.g. a full week of work) would make it possible to have reliable measures of the difficulty or inadequacy of the work environment to the required task and would give the ergonomist the opportunity to make a judgment based on a solid set of observations.

However, even though human pose estimation is a well-known problem in computer vision, it remains a complex problem whose existing solutions do not meet the

needs of this thesis.

On the one hand, we must distinguish between 2D human pose estimation, where the coordinates of the joints of the human body are given as corresponding pixels in the image, and 3D human pose estimation where the coordinates are data in a real world coordinate system. Today, 2D human pose estimation is a nearly solved problem for images in a wide range of different scenarios. However, since joint angles are determining factors for working conditions in the context of ergonomics, 2-dimensional analysis does not make it possible to recognize problematic postures. Human pose estimation in 3D is a much more complex problem because this is intrinsically an ill-posed problem in the case of a single image source. In reality, there is a natural ambiguity in estimating depth from a single image (like a hand leaning forward or backward can have the same 2D projection on the image).

The second major limitation is that these methods require training labels associated with the predictions (supervised learning) and for which it was necessary to set up a very complex recording system (for example a motion capture system with markers, see Figure 1.3).

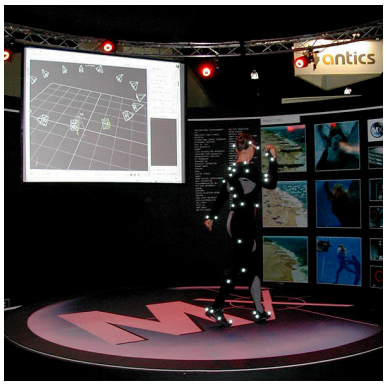

Figure 1.3: "A dancer wearing a suit used in an optical motion capture system." Light dots are the markers installed on her cloth to captured the coordinates, allowing the synthetic reconstruct in the computer space shown in the back. Image source: [wikipedia, 2003b] Motion capture

Therefore, the most recent data is only captured in a very constrained environment. However, we are more interested in an unconstrained framework because we want to perform analysis while avoiding the background impact of working conditions as much as possible. The neural network model must therefore be able to adapt to this environment without constraint.

To obtain an estimation of the 3D human pose with high precision, we therefore propose to decompose the whole task into several steps and to combine the strengths of each of them individually. First of all, we propose to perform a 2D human pose estimation from an image, because it is already well performing in an unconstrained environment. Then, starting from a 2D human pose, we take it to 3D with another neural network. Since this step only uses keypoint coordinates rather than images, the environment is no longer a performance-degrading factor. To alleviate the ambiguity problem, we propose using the joint distribution prior or using a more complex model, both of which can increase the prediction accuracy.

In the following chapters, we first make a brief explanation of the related works

in Chapter 2, as well as the state of the art under different categories of estimation of the human pose. Then, in chapter 3, we explain our problem in detail, as well as our general methodology. From chapter 4 to chapter 7, we present our 3 different research projects, including **Decanus to Legatus: Synthetic training for 2D-3D human pose lifting** [Zhu and Picard, 2022], which proposes an algorithm that generates synthetic 3D human skeletons on the fly during training to create new poses outside of common datasets, **H3WB: Human3.6M 3D WholeBody Dataset and Benchmark** [Zhu et al., 2023b], which proposes a framework for creating full-body 3D annotations from multi-view images along with a few datasets and a benchmark built based on this framework to increase the capacity of the skeleton model, and **PIUS: Pose Interpolation at extremely low and Uneven framerate** [Zhu et al., 2024] which proposes an algorithm that models human movement as a continuous function over time to deal with total human occlusion by obstacles in video sequences, as well as a real-time demonstration working on a CPU-only computer. In chapter 8 we give our conclusion.

Chapter 2

Related works

In this chapter, we provide a brief introduction to how ergonomic studies have been combined with deep learning as well as the deep neural network structures commonly used for human pose estimation, the state-of-the-art methods developed for human pose estimation, and several less developed topics on human poses that are related to our works.

2.1 Background

2.1.1 Ergonomic and deep learning

In recent years, the study of ergonomics has been greatly improved with the help of computer algorithms, and then extended to deep learning methods. Two great examples of domain interaction between ergonomics and deep learning are human-machine interaction and the study of musculoskeletal disorders.

Applying deep learning to human-machine interaction for ergonomics is a natural idea. The machine must improve its algorithm to adjust its positioning for user comfort, with deep learning methods being a very good choice for observing and optimizing these parameters. For example, Gholami et al.[1] proposed a framework to analyze both usability of online measurements of human body configurations, applied on a teleoperated robot with the Mocap 3D mouse system as user interface. Yazdani et al.[2] proposed Differentiable Upper Limb Assessment (DULA) to predict risk assessment from 10 upper body keypoints using a deep neural network, and maintain the same level of accuracy with the traditional rapid upper limb assessment (RULA) on the same task. They proposed in their later work[3] Differentiable Entire Body Assessment (DEBA) which extend DULA functionality to entire body. Based on DULA system[4], they introduced a framework allowing robotic arms to perform teleoperations with postural estimation and optimization. Cvetkovic et al.[5] studied the effect of random vibrations on the human body by having 35 experiment participants sit in a car seat with disturbance, showing that the direction of the disturbance and the kinematics of the body segments are the main factors leading to the peak translations of the signal. Shafti et al.[6] used an RGB-D camera to capture human posture

[1] [Gholami et al., 2021]

Quantitative physical ergonomics assessment of teleoperation interfaces

[2] [Yazdani et al., 2021a]

DULA: A differentiable ergonomics model for postural optimization in physical HRI.

[3] [Yazdani et al., 2022]

Differentiable ergonomic risk models for postural assessment and optimization in ergonomically intelligent pHRI

[4] [Yazdani et al., 2021b]

Ergonomically intelligent physical human-robot interac- tion: Postural estimation, assessment, and optimization

[5] [Cvetković et al., 2023]

Explaining human body responses in random vibration: Effect of motion direction, sitting posture, and anthropometry

[6] [Shafti et al., 2018]

Real-time robot-assisted ergonomics

and optimize the robot's parameters to maintain optimal ergonomics for humans. All these works show the great utility of using deep learning to control robot for better interaction with humans.

On the other hand, musculoskeletal health[7], according to World Health Organisation, "refers to the performance of the locomotor system, comprising intact muscles, bones, joints and adjacent connective tissues. Musculoskeletal conditions are typically characterized by pain (often persistent) and limitations in mobility and dexterity, reducing people's ability to work and participate in society." The use of deep learning methods to study work-related musculoskeletal disorders (WMSD), or work-related body fatigue for short, in ergonomics is also very common, which is also the aim of study of this thesis. Such a study must be quantified with numbers to allow researchers to analyze the observed human. There are two main ways to obtain this digital data, including sensor-based methods such as motion capture (Mocap), inertial measurement units (IMU), electromyogram (EMG) sensor which places sensors on the human body to directly measure data, as well as vision-based methods by capturing images or videos and using deep learning algorithms to obtain the necessary data. Normally, sensor-based methods are more accurate, while vision-based methods are cheaper and easier to generalize.

[7] [WHO, 2022]

Musculoskeletal-conditions

Some examples of sensor based methods are: Ma et al.[8] proposed a framework in 2011 to use Mocap to capture real human motion, then combining with a given digital human model as well as a physics engine, their system allows the simulation of human motion in a virtual physical environment and mainly studied the case of muscle fatigue. Lorenzini et al.[9] proposed to study WMSDs using an online approach to monitor kinematic and dynamic quantities on workers. These quantities are based on the positions and orientations of each individual joint relative to the pelvis joint. The proposed framework is then studied under 3 different actions including object lifting/lowering, drilling and painting with a tool. Mudiyanselage et al.[10] evaluated the ability of a surface EMG-based system to detect harmful body movements during material handling. 4 different machine learning methods were compared and showed that decision tree beats support-vector machine, K-nearest neighbor and random forest in terms of harm risk prediction accuracy with a small margin. Gelaw et

[8] [Ma et al., 2011]

A framework of motion capture system based human behaviours simulation for ergonomic analysis

[9] [Lorenzini et al., 2021]

An online multi-index approach to human ergonomics assessment in the work place

[10] [Mudiyanselage et al., 2021]

Automated workers ergonomic risk assessment in manual material handling using semg wearable sensors and machine learning

al.[11] studied WMSDs caused by prolonged sitting in a chair with bad postures using a chair with pressure sensors in the backrest. The data is separated according to a controlled and realistic scenario and performed classification tasks with 5 different machine learning algorithms, the best of which achieve 98% and 97% accuracy, respectively. These show that sensor based methods are able to obtain very precise data for accuracte analysis.

Some examples of vision-based methods are: Skorvankova et al.[12] proposed a synthetic dataset of 100,000 annotations with binary silhouette images, grayscale images, 3D skeletons, 3D surface point cloud and 3D mesh with 16 anthropometric body measurements, as well as 2 methods basic with predicted 16 values from silhouettes or a point cloud. Bayat et al.[13] proposed to use a convolutional neural network to predict the shape of the 3D segment of the spine from 2D images and a centroid point indicating the segment to be predicted. Marcon et al.[14] (see Figure 2.1) proposed to study WMSD of dentist during surgery by sticking several image markers on the dentist's back as a tracking pattern to construct their 3D posture with computer vision algorithms. Olivas-Padilla et al.[15] proposed to deal with WMSD by optimizing human-robot collaboration through the use of motion data and posture recognition as well as spatial adaptation which are believed to reduce the user's ergonomic risk. The hypotheses are tested in a television manufacturing process scenario, measured by two performance indicators: the percentage of spatial adaptation of the robot used and the amount of human operation effort is reduced. These vision-based methods prove the capability of computer vision method to achieve very good results even without using specific equipment and sensors, allowing stronger generality to practical uses.

In our case, in order to maintain generality and without the support of hardware sensors, we choose to use vision-based methods.

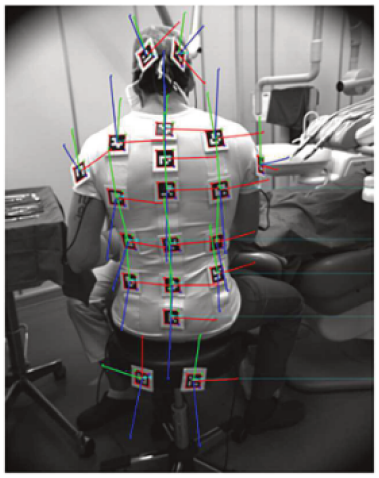

Figure 2.1: Marcon places stickers of different patterns on the dentist's back during surgery to track body movements, which should be less expensive than using motion capture systems. Image source: [Marcon et al., 2017] Postural assessment in dentistry based on multiple markers tracking

2.1.2 Deep learning network structures for human pose estimation

According to Tom Mitchell[16], the definition of machine learning is that, "A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves

[11] [Gelaw and Hagos, 2022]

Posture prediction for healthy sitting using a smart chair

[12] [Skorvánková et al., 2021]

Automatic estimation of anthropometric human body measure ments

[13] [Bayat et al., 2021]

Inferring the 3d standing spine posture from 2d radiographs

[14]

[15] [Olivas-Padilla et al., 2022]

Computational ergonomics for task delegation in human-robot collaboration: spatiotemporal adaptation of the robot to the human through contactless gesture recognition

[16] [Mitchell, 1997]

Machine Learning

with experience E."

In practice, the entire learning procedure can be generally formulated as an approximation problem with a model represented as a function $\mathscr{F}$ follows the equation $y = \mathscr{F}(x)$ with $x \in X$ is the collection of input data, and $y \in Y$ is the collection of the expected output data, called groundtruth. We want to improve $\mathscr{F}$ through learning so that the predicted value from learnt $\mathscr{F}(x)$ should be as close to the groundtruth y as possible. The size, shape and range of values of x or y can be varied depending on the task.

It is the same case for deep learning, but $\mathscr{F}$ is now a neural network with learnable parameters. Thus, choosing an appropriate form of network structure for $\mathscr{F}$ is an important factor for the performance of any method. Here we briefly introduce common network structures dealing with different input data structures.

Layers

A deep learning network is normally composed of several structures called layers, with each layer taking as input the data produced by the previous layer, calculating some type of transformation, and passing its output to the next layer. Layers can be separated into two types: linear and non-linear layers. Since two successive linear operations are equivalent to a single linear operation, alternating between linear and nonlinear layers ensures that a subpart of the network is not equivalent to a large linear layer, allowing the network to approximate much larger and more difficult functions.

A commonly used linear layer is **Fully connected layer** (FC layer), in which each value of the input of this layer is connected to each value of the output of this layer. The FC layer can be used on any type and dimension of vector data, allowing the transformation of data into any form we need. The disadvantage is that the number of parameters to learn is the product of the dimension in the input vector and the dimension in the output vector, which makes the calculation expensive in terms of number of operations when the data size is large.

Another commonly used linear layer is **Convolutional Layer** (Conv Layer) whose input vector is convolved with a learned kernel. The Conv Layer requires significantly fewer parameters to learn than a typical FC layer and also maintains the translation

invariance, which makes the Conv Layer a good choice for processing images.

The nonlinear layer is crucial for neural networks because it allows the network to learn very complex data. Commonly used nonlinear layers are pooling layers (e.g. maxpooling) and activation functions after each linear layer. (e.x. Relu, Sigmoid, Tanh, Softmax, etc.)

MLP

A neural network formed by a group of FC layers and non-linear activation functions is called Multi-Layer Perceptron (MLP). With the ability to have an arbitrary number of input and output dimensions, as well as an arbitrary number of possible layers, it is widely used for 2D to 3D human pose lifting tasks for pose estimation human, because the input and output data are often coordinate vectors.

CNN

A neural network formed by a sequence of convolutional layers and activation functions is called a convolutional neural network (CNN), even if there are a few fully connected layers at the end to adjust to the necessary output shape. As mentioned earlier, convolutional neural networks are one of the state-of-the-art model for processing images, so they are widely used for human pose estimation from images or videos.

Here lists some commonly used convolutional neural networks: VGG[17] is a typical pyramid-like CNN structure with convolutional layers at the beginning and two fully connected layers at the end for the classification task. Googlenet[18], or InceptionNet, introduced the perception block, allowing multiple convolutions or poolings applied in the same layer and the results are concatenated so that the network can obtain information from a shallow neighborhood to a broader neighborhood. Resnet introduced residual connection (see Figure 2.2), so that each block only needs to learn the amount of residual value changed between input and output, allowing to deal with vanishing or exploding gradient problems in traditional deep networks (i.e., when the gradients of the loss function become too small or too large, the network either stops learning or diverges. This allows the use of much deeper networks with hundreds

[17] [Simonyan and Zisserman, 2015]

Very deep convolutional networks for large-scale image recognition

[18] [Szegedy et al., 2015]

Going deeper with convolutions

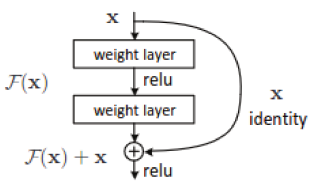

Figure 2.2: Residual module allows network really go 'deep' with hundreds of layers. Image source: [He et al., 2015] Deep residual learning for image recognition

of layers. Densenet[19] add a connection between every two residual Resnet blocks, making it a dense connection. There are many other different CNN structures with similar overall structure but small variations in details. All of these models are a good option to process image data, as well as task of estimation of human pose from an image.

[19] [Huang et al., 2017]
Densely connected convolutional networks

Transformer (in bold like MLP)

Transformer[20] is first introduced in 2017, initially for natural language processing tasks, but its structure is suitable for many other areas, including human pose estimation. Right in front of an MLP in each transformer block, the authors placed an attention module (see Figure 2.3) which is a nonlinear transformation with learnable

[20] [Vaswani et al., 2017]
Attention is all you need

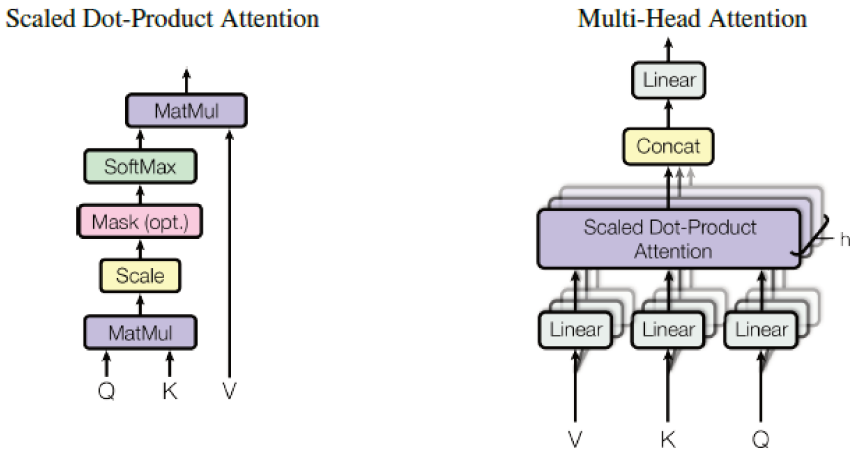

Figure 2.3: Attention module from Transformer. Image source: [Vaswani et al., 2017] Attention is all you need

parameters allowing the layer to choose important values of the input to pay more attention to and to ignore values of the input that are irrelevant to the task. As transformers also deal with vector data and the attention mechanism can handle missing or hidden values well, it is also a perfect framework for 3D human pose estimation with vector data like 2D and 3D coordinates, even with missing values due to occlusions.

Vision Transformer (ViT)[21] is a variant of transformer to manage images. It separates an input image into small patches of 16×16 pixels and they are vectorized and fed to a typical transformer encoder. In this way, it allows processing images with fully connected layers instead of classical convolutional layers, and becomes a

[21] [Dosovitskiy et al., 2021]
An image is worth 16x16 words: Transformers for image recognition at scale

choice of network structure for human pose estimation from an image or video.

RNN, LSTM and GRU

A neural network that recurrently pass the output from some neurons as part of the subsequent input to the same neurons is called a recurrent neural network (RNN). This "memory" capability makes RNN a popular choice for processing sequential data such as time series and text. However, the 'memorized' values will rapidly been ignored with smaller and smaller factor after more iterations , which is called vanishing gradient problem, makes it difficult for the network to have a long memory.

To address this problem, a special type of RNN, called long short-term memories (LSTM)[22], is introduced, which uses gates to control which values to keep (cell state), which value to forget (forgotten gate), which value to learn (input gate) as well as the values to return (output gate). The cell state is the long term memory and the combination of the other 3 gates forms the short term memory. However, LSTM has a rather complicated structure and that is why a gated recurrent unit (GRU)[23] is proposed, which is cheaper to compute while providing similar performance to LSTM. It replaced the multiple gates in LSTM by only reset-gate and update-gate, thereby reducing the number of calculations.

These RNN structures, due to their ability to process sequential data, become good choices for processing videos or sequences of human poses.

[22] [Hochreiter and Schmidhuber, 1997]

Long Short-Term Memory

[23] [Cho et al., 2014]

Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation

2.2 3D Human pose estimation

2.2.1 3D human pose estimation from image

In recent years, monocular 3D human pose estimation has been widely explored in the community. The models can be mainly categorized into generative models which fit 3D parametric models to the image, and discriminative models which directly learn 3D keypoint positions from images . Generative models try to fit the shape of the entire body and as such are great for augmented reality or animation purpose. However, they tend to be less precise than discriminative models. On the other hand, a difficulty that the discriminative models have is that depth information is hard to infer

from a single image when it is not explicitly modeled by body constraints, and thus additional bias must be learned using 3D supervision, multiview spatial consistency or temporal consistency.

Discriminative models can also be categorized into one stage models which predict directly 3D poses from images and two stage methods which first learn a 2D pose estimator, then lift the obtained 2D poses to 3D.

One stage methods

The problem of one-stage 3D human pose estimation from an image can be formally defined as follows: given an RGB image, the network must directly predict the 3D poses of humans in the image without intermediate 2D supervision.

Since one-stage methods are end-to-end, they don't have intermediate supervisions and more often requires data that already contains 3D information for supervision, such as multiview images. For example, Mehrizi et al.[24] used multiview data to enhance the 3D human pose estimation result. Rhodin et al.[25] proposed to use multiview-consistent 2D reprojection supervision and few shot 3D label supervision, together with a regularization to penalizes predictions that drift too far away from the initial prediction during training. Luvison et al.[26] proposes to estimate 3D coordinates in absolute values as well as a consensus-based optimization algorithm to estimate the unknown camera intrinsic and extrinsic parameters for reprojection consistency. Sengupta et al.[27] takes as input a group of images of the same person but without pose or camera constraints, optimized jointly with the SMPL model using a probabilistic pose and shape model. Pavlakos et al.[28] propose a fine discretization of 3D space and prediction of per-voxel probabilities for each joint, as well as a coarse-to-fine prediction scheme that allows iterative refinement. Benzine et al.[29] proposed a top-down method predicting human bounding boxes and 2D/3D poses with the help of pre-defined anchors which are the basic complete 3D poses from which the network practice refinement and avoid occlusion problems, and then also proposed a bottom up method[30][31] which simultaneously predicts 2D heatmaps and occlusions robust pose maps for 3D coordinates for each joint, and regroup joints into persons according to their embedding values for single shot multiperson prediction task.

[24] [Mehrizi et al., 2018]

Toward marker-free 3d pose estimation in lifting: A deep multi-view solution

[25] [Rhodin et al., 2018]

Learning monocular 3d human pose estimation from multi-view images

[26] [Luvizon et al., 2022]

Consensus-based Optimization for 3D Human Pose Estimation in Camera Coordinates

[27] [Sengupta et al., 2021]

Probabilistic 3d human shape and pose estimation from multiple unconstrained images in the wild

[28] [Pavlakos et al., 2017]

Coarse-to-fine volumetric prediction for single-image 3d human pose

[29] [Benzine et al., 2020]

PandaNet : Anchor-Based Single-Shot Multi-Person 3D Pose Estimation

[30] [Benzine et al., 2019]

Deep, Robust and Single Shot 3D Multi-Person Human Pose Estimation from Monocular Images

[31] [Benzine et al., 2021]

Single shot 3D multi-person human pose estimation in complex images

Due to the difficulty of acquiring accurate 3D annotations, as well as limited datasets with controlled environments and specific camera settings, one-stage methods suffer more from generalibility than two-stage methods.

Two stage methods

2D human pose estimation from image The problematic of 2D human pose estimation from image can be formally defined as following: given an RGB image, the network needs to detect the 2D poses of human in the image. The representation model of human pose could be contour-based[32][33], volume-based[34] or skeleton-based[35]. Due to the necessity of our work to compute the joint angles, the skeleton-based model is the only one that meets our needs and, fortunately, it is also the most developed human pose representation model in the research community.

The most common state-of-the-art methods for 2D human pose estimation from an image can be separated into two categories: Top-down, in which the network first detects each individual human from the image, then for each individual human, the network detects the key points of the pose, and the bottom-up framework , in which the network first detects all the keypoints that existed inside the images, and then clusters the keypoints to form individual humans[36][37]. Both frameworks have achieved very promising results, as well as the existence of the larger quantity and diversity of 2D datasets like COCO dataset [38](see Figure 2.4), MPII dataset[39], etc.

Figure 2.4: COCO dataset provides a wide range of indoor and outdoor scenario of image data with human within. Image source: [cocodataset, 2016]

allowing us to directly adapt these existing models and focus more on the harder part:

[32] [Ju et al., 1996]

Cardboard people: A parameterized model of articulated image motion

[33] [Kuehne and Woerner, 2010]

Motion Segmentation of Articulated Structures by Integration of Visula Perception Criteria

[34] [Sidenbladh et al., 2000]

A framework for modeling the appearance of 3d articulated figures

[35] [Felzenszwalb and Huttenlocher, 2005]

Pictorial structures for object recognition

[36] [Pishchulin et al., 2016]

DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation

[37] [Insafutdinov et al., 2016]

DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model

[38] [Lin et al., 2014]

Microsoft coco: Common objects in context

[39] [Andriluka et al., 2014]

2d human pose estimation: New benchmark and state of the art analysis

the 3D human pose.

3D human pose estimation from 2D human pose The problem of lifting 2D human poses to 3D can be formally defined as follows: Given a 2D human pose, the network must predict the corresponding 3D poses.

Lifting 2D pose to 3D is somewhat of an ill-posed problem because of depth ambiguity. But the larger quantity and diversity of 2D datasets as well as the much better performance already obtained in 2D human pose estimation provide a strong argument for many researchers, including us, to focus on the lifting of 2D human poses to 3D. One simple baseline of such task is done by Martinez et al.[40], who use a small MLP to realize 2D to 3D human pose lifting (see Figure 2.5).

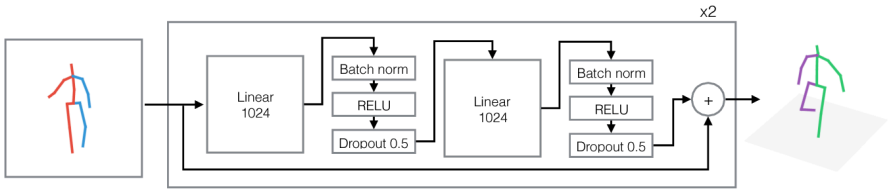

Figure 2.5: A very simple 4 layer MLP doing 2D-3D lifting task. Image source: [Martinez et al., 2017c] A simple yet effective baseline for 3d human pose estimation

One classic method to help improve lifting performance is to use the human body's built-in priors that are commonly agreed upon, such as the order of the human body's kinematic tree as well as left-right symmetry. For example, Chen et al.[41] proposed combining bone length and camera parameters with 2D coordinates as input, as well as adding a direction loss for each human branch. Park et al.[42] divide the human body into 5 parts and extract the features of all combination of pairs of two of these parts. Biswas et al.[43] proposed to use the same lifting network architecture to back-project the estimated 3D pose into 2D to allow for weak supervision, as well as the need to combine the symmetric bone length constraint. Hardy et al.[44] argue that, in an unsupervised adversarial lifting task, using an independent two-branched model of the torso and legs is the best 2D representation to learn. Wei et al.[45] based on the baseline method, proposed to add a view-invariant hierarchical correction network that transforms both the predicted 3D pose and the ground truth into a fixed view, and judged with a discriminator. Mehta et al.[46] transferring features from 2D pose

[40] [Martinez et al., 2017c]

A simple yet effective baseline for 3d human pose estimation

[41] [Chen et al., 2021]

Estimation of 3d human pose using prior knowledge

[42] [Park and Kwak, 2018]

3d human pose estimation with relational networks

[43] [Biswas et al., 2019]

Lifting 2d human pose to 3d

[44] [Hardy et al., 2022]

Optimising 2d pose representation: Improve accuracy, stability and generalisability within unsupervised 2d-3d human pose estimation

[45] [Wei et al., 2019]

View invariant 3d human pose estimation

[46] [Mehta et al., 2017]

Monocular 3d human pose estimation in the wild using improved CNN supervision

estimation to 3D pose estimation task, as well as multi-model 3D pose prediction and fusion.

Another common practice is to find another representation of the data instead of joint coordinates in order to facilitate learning. For example, Noguer et al.[47] proposed using a distance matrix between joints to represent both 2D and 3D human pose instead of coordinates, as well as using an MDS algorithm to convert the distance matrix back to 3D human pose coordinates. Kang et al.[48] proposed grid convolution. By using a binary assignment matrix that maps the graph pose to a grid pose, it allows the network to perform convolution to raise the 2D pose to 3D, which is normally unreasonable with just coordinate values. Zhang et al.[49] proposed to keep the contextual information extracted from the 2D pose estimation network to the lifting network, as well as learn the 3D position heatmap from the 2D heatmaps and context information. Krishna et al.[50] proposed using quaternions to represent 3D human pose in order to avoid the constraints of bone lengths with 3D coordinates or discontinuities and singularities with Eular angles or axis-angles. Xu et al.[51] propose graph stacked hourglass networks that estimate human skeleton models from a complete skeleton to a simplified 4-joint skeleton, performing coarse-to-fine estimation through intermediate layers. Wandt et al.[52] propose to estimate the 3D human pose under the canonical space and the camera which transforms from the canonical space to the camera space corresponding to the input image.

In addition to using a network to learn to lift, some researchers use a predefined model to adapt the 3D pose and shape to the image as well as 2D pose. SMPL[53] is one of the recent most popular model and its fitting algorithm SMPLify from Bogo et al.[54] (see Figure 2.6), which uses a CNN to predict the location of 2D joints and then fits a 3D parametric body model to estimate the shape and pose of the 3D body by minimizing the error between the projected 3D model joints and the detected 2D joints. Based on this, Kissos et al.[55] proposed to use a full perspective projection camera model instead of weak perspective camera model in SPIN algorithm [56] and combine it with SMPLify algorithm to regress directly from image to 3D pose and shape.

There are also researchers which deal with multiview 2D poses, like Chen et al. [57]

[47] [Moreno-Noguer, 2016]

3d human pose estimation from a single image via distance matrix regression

[48] [Kang et al., 2023]

3d human pose lifting with grid convolution

[49] [Zhang et al., 2021]

Deep monocular 3d human pose estimation via cascaded dimension-lifting

[50] [Krishna et al., 2021]

Signpose sign language animation through 3d pose lifting

[51] [Xu and Takano, 2021]

Graph stacked hourglass networks for 3d human pose estimation

[52] [Wandt et al., 2021]

Canonpose: Self-supervised monocular 3d human pose estimation in the wild

[53] [Loper et al., 2015a]

SMPL: A skinned multi-person linear model

[54] [Bogo et al., 2016]

Keep it SMPL: automatic estimation of 3d human pose and shape from a single image

[55] [Kissos et al., 2020]

Beyond weak perspective for monocular 3d human pose estimation

[56] [Kolotouros et al., 2019]

Learning to reconstruct 3d human pose and shape via model-fitting in the loop

[57] [Chen and Ramanan, 2017]

3d human pose estimation = 2d pose estimation + matching

Figure 2.6: 3D pose and shape rendered on 2d image with SMPLify algorithm. Image source: [Bogo et al., 2016] Keep it SMPL: automatic estimation of 3d human pose and shape from a single image

prepared a large library of 3D poses and used the nearest neighbor model to find the closest pair between a given 2D pose and the projection of a 3D pose from the library with virtual cameras. Schwarcz et al. [58] aim to optimize a conditional random field of 3D joint reconstruction of several people from neighboring frames of 2D multi-view video in order to reduce noise and ambiguity. The matching between spatial images is carried out with the IoU information of the bounding box of each individual, and the matching between temporal images is carried out with the smallest average distance between each ray extending in 3D. Qiu et al.[59] performs cross-view feature fusion on 2D pose estimation on multi-view images, where the 3D pose estimation is performed by dividing the 3D space into grids called pictorial structure model and minimizing the projection error on 2D heat maps. Dong et al.[60] used a multi-way matching algorithm to cluster the multiple 2D human poses detected in all views to ensure a consistent matching, and then also infer a 3D pose with pictorial structure. Chen et al.[61] iteratively refined 3D pose using triangulation from multi-view 2D poses, camera parameters, bone length constraints, and structural information. Iqbal et al.[62] proposed using 2.5D estimation by predicting 2D coordinates and depth, as well as rigid alignment of 3D predictions from multiple views. Since the calibration of multi-view cameras is still complicated in practice in the wild, these methods do not fit our framework.

[58] [Schwarcz and Pollard, 2019]

3d human pose estimation from deep multi-view 2d pose

[59] [Qiu et al., 2019]

Cross view fusion for 3d human pose estimation

[60] [Dong et al., 2019]

Fast and robust multi-person 3d pose estimation from multiple views

[61] [Chen et al., 2022]

Structural triangulation: A closed-form solution to constrained 3d human pose estimation

[62] [Iqbal et al., 2020]

Weakly- supervised 3d human pose learning via multi-view images in the wild

2.2.2 3D human pose estimation with temporal information

The problem of 3D human pose estimation from video can be formally defined as follows: given a sequence of RGB images covering certain time periods, the network

needs to predict 3D poses for one or all images.

Compared to 3D human pose estimation from a single image, this task contains additional information about the images before and after the one to be predicted, allowing to improve the prediction accuracy using temporal coherence across time periods assuming the human cannot move a lot in a very short time (typically $\leq$ 1s). For example, Choi et al.[63] proposed to separately predict the pose of the current frame from past features, future features, and all features, and then integrate the predictions to improve the temporal consistency of pose and shape. Kanazawa et al.[64] proposed to predict both a past pose and a future pose as well as a current pose with a adversarial prior loss to ensure the validity of the predictions, as well as a hallicinator that attempts to learn the same feature but only from a single current image. Sometimes this temporal consistency allows some self-supervision between successive data, allowing learning without the entire sequence or without 3D annotations. Such methods may be more generalizable to many more data sequences. For example, Takahashi et al. [65] process videos from unsynchronized and uncalibrated cameras using a relaxed reprojection error based on a confidence map, and jointly optimize the temporal offset between videos, camera parameters, and 3D poses. Li et al.[66] proposed to use a 3D trajectory optimization algorithm on a 3D pose sequence predicted from a pre-trained network to create a pseudo-supervision label. Einfalt et al.[67] use a transformer network and padding a sparse 2D pose sequence with a learnable upsampling token to realize a monocular temporally sparse 2D pose sequence up to a temporally dense 3D pose estimation. Luvison et al.[68] jointly estimate 2D/3D poses and action recognition from a pose sequence with a multi-task pyramid structure.

Due to the large variation in human position in the image sequence, some sort of data normalization is necessary to facilitate learning. For example, Tekin et al.[69] proposed using a CNN to move image windows to a centralized human in each image and then another CNN to regress the 3D pose from the concatenated 3D features. Wang et al.[70] proposed a motion loss based on the movement of points centered on a root joint. Zell et al.[71] learn new 3D poses from linear combinations of a set of base poses learned by principle component analysis (PCA) on motion sequences.

63 [Choi et al., 2021]

Beyond static features for temporally consistent 3d human pose and shape from a video

64 [Kanazawa et al., 2019]

Learning 3d human dynamics from video

65 [Takahashi et al., 2018]

Human pose as calibration pattern; 3d human pose estimation with multiple unsynchronized and uncalibrated cameras

66 [Li et al., 2019]

On boosting single-frame 3d human pose estimation via monocular videos

67 [Einfalt et al., 2022]

Uplift and upsample: Efficient 3d human pose estimation with uplifting transformers

68 [Luvizon et al., 2020]

Multi-task Deep Learning for Real-Time 3D Human Pose Estimation and Action Recognition

69 [Tekin et al., 2015]

Direct prediction of 3d body poses from motion compensated sequences

70 [Wang et al., 2020]

Motion guided 3d pose estimation from videos

71 [Zell et al., 2017]

Joint 3d human motion capture and physical analysis from monocular videos

Zhou et al.[72] also decompose the poses into linear combinations of basis poses and optimize the 3D pose via an expectation-maximization algorithm based on 2D joint uncertainties.

[72] [Zhou et al., 2016]

Sparseness meets deepness: 3d human pose estimation from monocular video

2.2.3 3D human motion prediction

The problem of 3D human motion prediction can be formally defined as follows: given a subset of 3D human motion in a sequence, the network must predict 3D human motion in the entire sequence. If the given subset consists of the frames at the start of the sequence, the task can be called future motion prediction, and if the given subset contains individual non-successive frames from the sequence, the task can be called motion interpolation, motion completion or motion infilling. (see Figure 2.7).

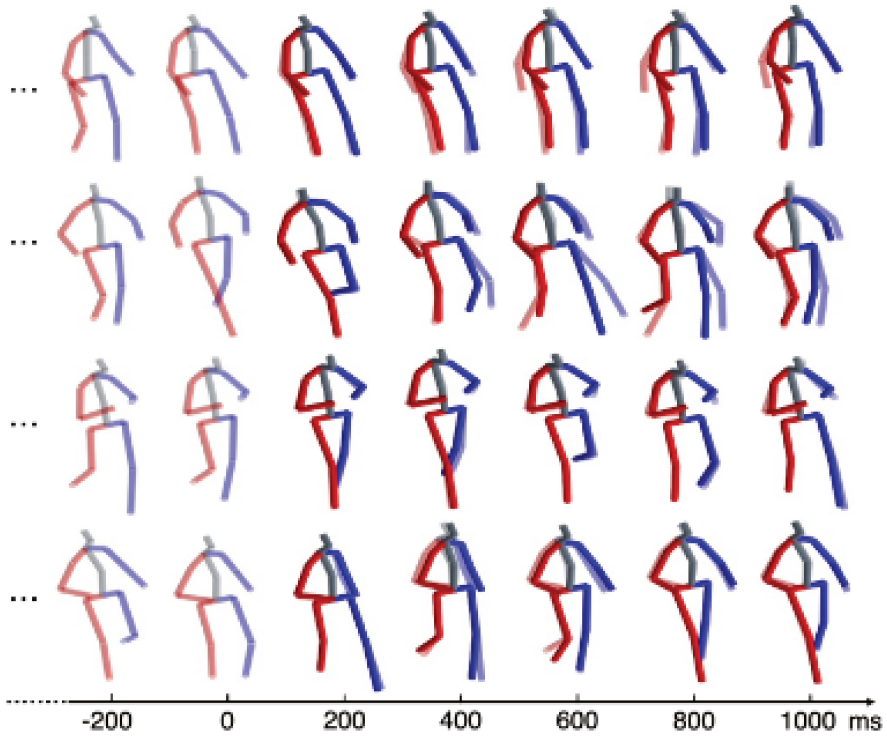

Figure 2.7: Example of human movement represented as a skeleton based on key points. A clear pattern of action can be observed over time. Image source: [Guo et al., 2022] Back to mlp: A simple baseline for human motion prediction

Compared to pose prediction from video, this task does not contain any data corresponding to the same time period that the pose is to be predicted, making prediction more difficult due to the lack of references and great uncertainty of human movement through time. Thus, temporal and spatial coherence become important factors in this

task. For example, Martinez et al.[73] proposed to use a sequence-to-sequence RNN model, which feeds the ground-truth to the encoder, and the error is calculated on the decoder which feeds its own predictions. Zhu et al.[74] proposed dynamics maps that encode spatial or spatio-temporal motion patterns as environmental features for long-term multi-model motion prediction. Mao et al.[75] proposed to use graph convolutional networks along with a discrete cosine transform to train in the trajectory space to avoid the temporal convolution filter. Sun et al.[76] proposed to iteratively refine the motion prediction between pose space and frequency space with a discrete cosine transform. Guo et al.[77] proposed to use MLP structure along with discrete cosine transform to transform data from coordinate space to trajectory space. Katircioglu et al.[78] introduces a pairwise attention mechanism to model the mutual dependencies of two objects to reason about the interactions, so as to apply the pose estimation of two dancing people. Yasar et al.[79] proposed to follow a curriculum learning by first learning the rough topological organization of the human body and then adjusting for accurate prediction.

In addition to temporal and spatial coherence, some works take advantage of the context of each movement to facilitate network learning. For example, Corona et al.[80] proposed to use RNN to simultaneously perform motion prediction and context understanding, which is achieved by using past object position, class, and human joints to predict interactions and context. Tanke et al.[81] predict symbolic labels to represent human intention, to facilitate the human motion prediction. Kiciroglu et al.[82] suggest predicting only a few key poses, which are normally the turning pose on the long term, such that interpolating the intermediate frames is sufficient for more accuracy. Sun et al.[83] used LSTM as the backbone of the prediction network and reinforced with an action classifier as well as an action memory bank to store the movement dynamics for each category.

Some researchers also take into account the uncertainty of motion human and use variational model or noises to simulate them to improve performance. For example, Gopalakrishnan et al.[84] used a Verso-Time Label Noise RNN model that can learn the noise process and future motion, as well as a loss of derived information. Aliakbarian et al.[85] proposed to make stochastic motion prediction using the root of

[73] [Martinez et al., 2017a]

On human motion prediction using recurrent neural networks

[74] [Zhu et al., 2023a]

A data-efficient approach for long-term human motion prediction using maps of dynamics

[75] [Mao et al., 2019a]

Learning trajectory dependencies for human motion prediction

[76] [Sun and Chowdhary, 2023]

Towards accurate human motion prediction via iterative refinement

[77] [Guo et al., 2022]

Back to mlp: A simple baseline for human motion prediction

[78] [Katircioglu et al., 2021]

Dyadic human motion prediction

[79] [Yasar and Iqbal, 2021]

Improving human motion prediction through continual learning

[80] [Corona et al., 2020]

Context-aware human motion prediction

[81] [Tanke et al., 2019]

Human motion anticipation with symbolic label

[82] [Kiciroglu et al., 2020]

Long term motion prediction using keyposes

[83] [Sun et al., 2021]

Action-guided 3d human motion prediction

[84] [Gopalakrishnan et al., 2018]

A neural temporal model for human motion prediction

[85] [Aliakbarian et al., 2020]

A stochastic conditioning scheme for diverse human motion prediction

variations to add stochastic noise to past data. Cheng et al.[86] trained an offline RNN for motion prediction, and then adopts a recursive least-squares parameter adaptation algorithm for online parameter adaptation and uncertainty estimation. Xu et al.[87] extends the deterministic motion prediction network into a Bayesian network, enabling uncertainty calculation and avoiding forced dangerous actions of the robot when dealing with unseen motions. Ding et al.[88] proposed uncertainty-aware motion prediction network by predicting the mean and variance of the keypoint instead of the coordinates.

[86] [Cheng et al., 2018]
Human motion prediction using adaptable neural networks

[87] [Xu et al., 2021]
Probabilistic human motion prediction via A bayesian neural network

[88] [Ding and Yin, 2021]
Uncertainty-aware human motion prediction

2.2.4 Human pose synthesis and training

While previous works are mainly developed and validated on benchmark datasets consisting of real samples of human images and poses captured by camera or sensors, real datasets normally suffer from limitation of scenarios and contexts of different poses due to the finite number of data samples against diversity of human appearances and viewpoints in real life. Thus, some works propose to synthesize data that resembles a real image or human pose, but has greater variation and show that training with these synthetic datasets achieves as good or even better performance than only with real data. Synthetic training has long been a popular option for estimating human body pose in 3D [89].

[89] [Shotton et al., 2011]
Real-time human pose recognition in parts from single depth images

A simple way to synthesize new human pose data is to augment the data on existing real data with some kind of variations, such as cropping part of the image data or changing the value of the pose data. (see Figure 2.8). These data augmentations do not require a generator to learn the distributions and characteristics of different poses. For example, for image data, Noghre et al.[90] use a 2D human detector on a very high resolution image, then crops these areas semi-randomly to form new image data. Huang et al.[91] proposed using random erase and cutout for single-area information dropping, as well as hide-and-seek (random block mask) and grid mask for multi-area information dropping on training images for training sample augmentation. These augmentations modify the input images while the target pose label are unchanged.

[90] [Noghre et al., 2022]
Adg-pose: Automated dataset generation for real-world human pose estimation

[91] [Huang et al., 2020]
How to train your robust human pose estimator: Pay attention to the constraint cue

For pose data, however, ground-truth 3D label can also be augmented. For ex-

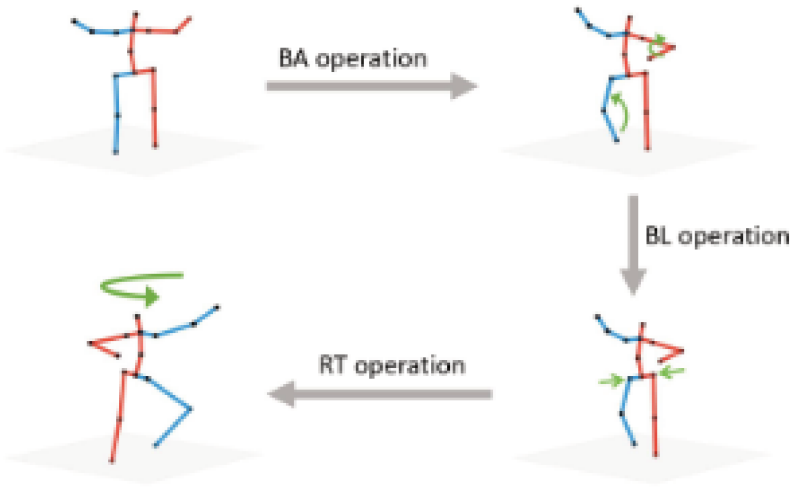

Figure 2.8: Cut-and-paste and adding noise is the most common way for data augmentation and new human pose synthesis. Image source: [Gong et al., 2021] Poseaug: A differentiable pose augmentation framework for 3d human pose estimation

ample, Gong et al.[92] propose to randomly adjust the bone length, bone angle, rigid viewpoint transformation, as well as a discriminator to ensure that the generation is a plausible pose. Jiang et al.[93] propose to use a pose transformation module that applies an affine transformation on the limbs as well as a pre-trained pose discriminator. Li et al.[94] perform data augmentation with crossover and mutation operations to evolve to possible new unseen poses.

Another way to synthesize human pose data is to use a generator to produce new poses after learning the distributions and features from real data. Although it is impossible to justify whether the samples generated from such a generator look real or not with per-pixel value supervision, because most of the samples produced are not part of the real dataset, a popular choice is to use a neural network called a discriminator, which takes real samples and generated fake samples as input to examine whether they are real or not. The combination of the generator network as well as the discriminator is called generative adversarial networks (GAN). During training, two networks are iteratively trained while freezing parameters of another one without doing back propagation, so they will both do better at their jobs in such a 'competitive' way. Such setup makes it possible to carry out many tasks without direct supervision from any sort of groundtruth data.

[92] [Gong et al., 2021]

Poseaug: A differentiable pose augmentation framework for 3d human pose estimation

[93] [Jiang et al., 2022]

Posetrans: A simple yet effective pose transformation augmentation for human pose estimation

[94] [Li et al., 2020]

Cascaded deep monocular 3d human pose estimation with evolutionary training data

Such a GAN structure was found to be available to generate both image data and pose data for the human pose estimation task. For example, for image data, Zhang et al.[95] generate a 2D human pose from only image captions describing human actions with a conditional GAN. Hukkelas et al.[96] used a GAN to synthesize a human figure from a masked image and a sparse set of keypoints, and the style is conditioned by the adaptive instance normalization in the generator. Roy et al.[97] synthesize new human poses with given images and text descriptions by first estimating a coarse representation of the pose from the text, then refining the estimate and the face as well as rendering the generated image by conditioning pose transfer. Chen et al.[98] use an unpaired post-guided GAN to synthesize a new pose from the initial image and a partial image indicating the parts of the pose (see Figure 2.9).

[95] [Zhang et al., 2020b]

Adversarial synthesis of human pose from text

[96] [Hukkelås and Lindseth, 2023]

Synthesizing anyone, anywhere, in any pose

[97] [Roy et al., 2022]

Tips: Text-induced pose synthesis

[98] [Chen et al., 2019]

Unpaired pose guided human image generation

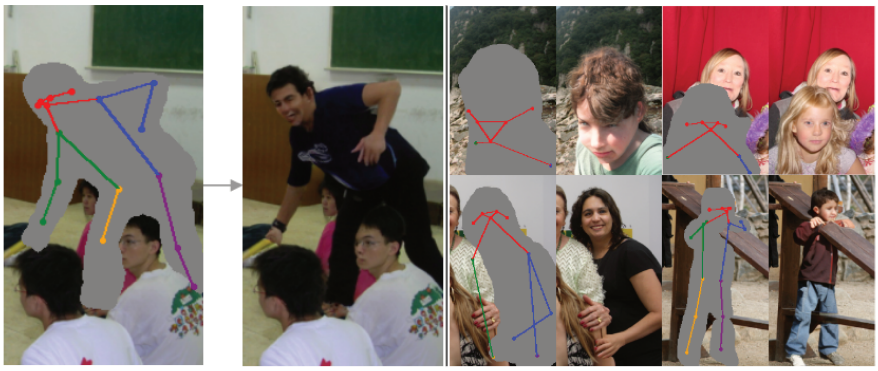

Figure 2.9: A GAN synthesizes realistic human figures with a masked image and a sparse set of keypoints. Image source: [Hukkelås and Lindseth, 2023] Synthesizing anyone, anywhere, in any pose

For pose data, Huang et al.[99] proposed to use a parametric model to analyze the skeleton generated by a GAN, ensuring the continuation of the skeletons generated via a video. Yang et al.[100] use pose sequence GAN to generate a 2D pose sequence from an input pose and a target action label. Some works learn a parametric model so that generators are seeded from a known distribution. For example, Xu et al.[101] learn a parametric model from a high-resolution 3D human scan and generate pose and shape by adjusting parameters. Chen et al.[102] sampled 3D pose from a dictionary, deforming the SCAPE model from a 3D pose to a 3D shape, then rendering the texture on the model with a variety of viewpoints and light sources, and finally setting in real image with background.

[99] [Huang et al., 2022]

Dh-aug: Dh forward kinematics model driven augmentation for 3d human pose estimation

[100] [Yang et al., 2018]

Pose guided human video generation

[101] [Xu et al., 2020]

Ghum amp; ghuml: Generative 3d human shape and articulated pose models

[102] [Wenzheng et al., 2016]

Synthesizing training images for boosting human 3d pose estimation

Another way to generate a new training image sample for 3D human pose estimation from image is the synthesis of another view, which given a person in one view as well as one pose in another view, the network generates the image of the same person in the given new view while keeping the texture alignment with the input image. For example, Zhang et al.[103] take an image, its input pose and a target pose. First they use an analysis generator to transfer the body part information with hierarchical deformation, then an image generator is used to synthesize the final image. Rochette et al.[104] synthesize new views of a human by practicing 3D pose estimation, then transfer the pose to a new view and render the Gaussian primitive model and appearance to synthesize the final image. Wu et al.[105] separately encodes an image and input/target poses with the transform module into style features and parser map, and merges to decode into a new human image in the target pose. Ma et al.[106] only needs an input image as well as a target pose, first roughly generates a human in the target pose and then refines it to fit the style examined by a discriminator. Tang et al.[107] divide the input and target pose into subgroups and calculate the local flow fields to see how the pose changes, then move on to the features extracted by the images for local deformation, and finally a global fusion of different parts is used to the final generated image. Balakrishnan et al.[108] take the image, input and target pose, then segment them into subgroups and uses spatial transformation to synthesize the foreground as well as the background obscured by the silhouette, synthesizes the new background to form the final generated image. Varol et al.[109] propose to estimate 3D human shape and augmenting motion, color and viewpoint to realize synthetic training.

Many of these methods also provide a synthetic dataset with their proposed method for generation to help the community, like Sminchisescu et al. [110] render synthetically generated poses on natural indoor and outdoor image backgrounds. Ghezelghieh et al. [111] use 3D graphics software and the CMU Mocap dataset to synthesize humans with different 3D poses and viewpoints. Pumarola et al. [112] created 3DPeople, a large-scale synthetic dataset of photorealistic images with a wide variety of human subjects, activities, and outfits. Varol et al.[113] propose to use sequences of 3D human motion capture data to render synthetically-generated but realistic images of people.

For our work, since 2D pose estimation already has rich collection of datasets, we

[103] [Zhang et al., 2020a]

Human pose transfer by adaptive hierarchical deformation

[104] [Rochette et al., 2021]

Human pose manipulation and novel view synthesis using differentiable rendering

[105] [Wu et al., 2022]

Pose guided human image synthesis with partially decoupled GAN

[106] [Ma et al., 2017]

Pose guided person image generation

[107] [Tang et al., 2021]

Structure-aware person image generation with pose decomposition and semantic correlation

[108] [Balakrishnan et al., 2018]

Synthesizing images of humans in unseen poses

[109] [Varol et al., 2021]

SURREACT: Synthetic Humans for Action Recognition from Unseen Viewpoints

[110] [Sminchisescu et al., 2006]

Learning joint top-down and bottom-up processes for 3d visual inference

[111] [Ghezelghieh et al., 2016]

Learning camera viewpoint using CNN to improve 3d body pose estimation

[112] [Pumarola et al., 2019]

3DPeople: Modeling the Geometry of Dressed Humans

[113] [Varol et al., 2017]

Learning from Synthetic Humans (SURREAL)

are only interested in generating realistic 3D poses as a set of keypoints in order to train a 3D lifting neural network. As such, we do not need to render visually realistic humans with meshes, textures and colors for this much simpler task.

2.3 Other human pose related topics

2.3.1 Human pose prior.

Due to the human body being highly constrained, it can be exploited as an inductive bias in pose estimation and pose synthesis, and is already widely used in recent research. For example, Bregleret al.[114] use a kinematic chain human pose model that follows skeletal structure, extended by Sigal et al.[115] with interpenetration constraints. Chow et al.[116] introduced the Chow-Liu tree, the maximum spanning tree of all the pairwise mutual information tree to model pairs of joints that exhibit high information flow. Lehrmannet al.[117] use a Chow-Liu tree that maximizes an entropy function based on nearest neighbor distances and learn local conditional distributions from data based on this tree structure. Akhter et al. [118] learn joint angle limits beforehand under local coordinate systems of 3 human body parts like torso, head and upper legs.

[114] [Bregler and Malik, 1998]

Tracking people with twists and exponential maps

[115] [Sigal et al., 2011]

Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation

[116] [Chow and Liu, 1968]

Approximating discrete probability distributions with dependence trees

[117] [Lehrmann et al., 2013]

A non-parametric bayesian network prior of human pose

[118] [Akhter and Black, 2015]

Pose-conditioned joint angle limits for 3d human pose reconstruction

2.3.2 Human wholebody

3D Body, hand and face pose estimation. While human pose estimation normally focuses mainly on the body and branches, there are also studies to estimate hand poses or facial expression. These two tasks are equally important because estimating hand pose can help study human-object interactions, while estimating facial expression allows the computer to study and analyze human emotions, thereby improving human-machine interaction performance. 3D hand pose estimation methods share similar approaches to body estimation methods, with one-stage and two-stage methods. First group of works estimate the hand pose from a single RGB image by directly regressing the key points of the 3D hand , mesh vertices , and parameters of parametric 3D hand models. Second group of works rely on intermediate 2D representations such as 2D keypoints and feature maps. Similarly, predominant 3D face pose estima-

tion methods regress the dense 3D face landmarks and face model parameters based on 3DMM [119].

[119] [Blanz and Vetter, 1999]

A morphable model for the synthesis of 3d faces

Figure 2.10: Wholebody means study body, face, hands and foot as a whole. Image source: [Jin et al., 2020] Whole-Body Human Pose Estimation in the Wild

3D Whole-body pose estimation. There are a several methods jointly estimating 3D whole-body pose. The first group of works are based on parametric human body models such as Adam model[120] and SMPL-X model [121]. MTC [122] is based on the Adam model, and first obtains 2.5D predictions, then optimizes the Adam model parameters. SMPLify-X optimizes the SMPL-X model parameter to fit 2D keypoints. Both methods achieved very impressive results with reasonable and accurate body and hand poses as well as clear facial expression, but the major drawback is that optimization-based methods are relatively slow and very sensitive to parameter initialization. Non-parametric methods follow different approaches to avoid heavy optimization procedure. DOPE[123] and FrankMocap[124] first train separate models of the body, hands and face. Then they combine these models into a learning framework. DOPE curates pseudo-ground truths from separate body models and uses these ground truths to supervise the distillation model. Similar to DOPE, ExPose[125] first obtains a pseudo-ground truth dataset by fitting the SMPL-X model on in-the-wild images, and trains a joint model to produce whole-body poses. All of these methods use many part-based datasets. Additionally, all of them produce different full body layouts with different number of full body key points. FrankMocap, DOPE and SMPLify-X estimate the whole body pose with 65, 139 and 144 keypoints respec-

[120] [Joo et al., 2018]

Total capture: A 3d deformation model for tracking faces, hands, and bodies

[121] [Pavlakos et al., 2019]

Expressive body capture: 3D hands, face, and body from a single image

[122] [Xiang et al., 2019]

Monocular total capture: Posing face, body, and hands in the wild

[123] [Weinzaepfel et al., 2020]

DOPE: distillation of part experts for whole- body 3d pose estimation in the wild

[124] [Rong et al., 2021]

Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration

[125] [Choutas et al., 2020]

Monocular expressive body regression through body-driven attention

tively.

2.3.3 Human pose completion

The problem of human pose completion can be formally defined as follows: given an incomplete human pose with missing values, the network must complete it by locating the missing keypoints. This task is very useful because, in reality, detected humans are often partially occluded by other objects, leading to incomplete human pose detection. To deal with this problem, Carissimi et al.[126] propose a network of variational denoising autoencoder to fill the missing key points in 2D pose completion. Bautembach et al. [127] select a small subset of poses from a database based on their distance from an incomplete 3D pose, and replaces missing keypoints with the corresponding averaged keypoints in the subset. Although being essential for real-world scenarios, pose completion has not been sufficiently explored.

[126] [Carissimi et al., 2018]

Filling the gaps: Predicting missing joints of human poses using denoising autoencoders

[127] [Bautembach et al., 2018]

Filling the joints: Completion and recovery of incomplete 3d human poses

2.3.4 Implicit Neural Representations of Human Motion

Implicit neural representations have gained significant attention following seminal works such as NeRF[128] and AtlasNet[129]. The core concept of implicit neural representations involves the use of a continuous function that maps spatial or temporal coordinates to represent a continuous surface. This notion has found applications in human motion, as seen in NeMF[130], which introduces a generative neural motion field parameterized in spatial-temporal space and express human poses depending on time and feature.

[128] [Mildenhall et al., 2020]

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

[129] [Groueix et al., 2018]

AtlasNet: A Papier-Mache Approach to Learning 3D Surface Generation

[130] [He et al., 2022a]

NeMF: Neural Motion Fields for Kinematic Animation

Chapter 3

General methodology

In this chapter, the technical objective of this entire thesis is detailed again. The general framework proposed to achieve the objective is presented, followed by the analysis of each intermediate part, as well as the difficulties, which lead to the work and contributions of this thesis in the following chapters.

3.1 Recall technical target

As briefly explained in the introduction, the technical objective of this thesis is to realise an autonomous analysis system which allows users to detect the risks of pain and potential injuries of people during their working time with only one photo. The analysis need to get the 3D relative positions of the limb joints of each person in the image, which allows calculating the angle of rotations in each direction, leading to the detection of over-twisted limbs whose angle is not in a comfortable zone.

3.2 System backbone

To achieve the objective of the thesis, the entire system is broken down into 3 consecutive steps which are applied one after the other (see Figure 3.1).

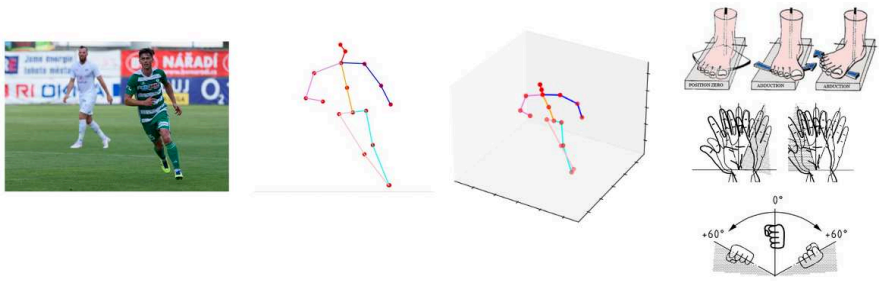

Figure 3.1: Here shows an example of how the results look before and after each step. Starting from a single image, 2D pose estimation, 2D to 3D pose lifting and geometric calculation will respectively provide the results shown in successive images. Image source: **soccer**: [Kreiss et al., 2021] OpenPifPaf:Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association, **foot/hand angles**: Template squelette 3D_v2_13012021.pptx from Ergonova Conseil.

1. First, the system will use a single 2D image captured by the camera as input and return the 2D coordinates of each human's joints in the image. This step correspondents to 2D human pose estimation from image.

2. Then, for each human in the image, the system will take a set of corresponding 2D coordinates as input and return their 3D coordinates in camera space. This step correspondents to 3D human pose estimation from 2D human pose.

3. Finally, for each set of 3D coordinates corresponding to the same individual, the system calculates the angles, as the length of limbs are fixed through time, which is not our concern. This step is called **Geometric computation**.

By combining all three steps, the system should be able to convert an image into pose and angle information. Even though each step has been extensively studied in the respective communities, there are still some difficulties that this work must resolve in order for the system to function properly. The biggest problems are listed below.

3.3 Problems to solve

3.3.1 Controlled environment vs 'In-The-Wild'

Although there is already a lot of promising work on monocular 3D human pose estimation, which can take a single 2D image as input and directly predict the 3D coordinates of human joints in 3D space, the reason for which we separate it into 2D pose estimation and 3D pose lifting is due to the high variation in environmental conditions that we need in this work. Our work requires the ability to be used in the working environment, which includes both indoor and outdoor scenarios, which can be called "in the wild", while most current state-of-the-art methods on monocular 3D humans Pose estimation is based on indoor scenes captured in a controlled environment with limited background type and predefined poses type due to hardware limitation. In order to obtain accurate human pose positions in 3D, a specific process called Motion Capture is applied to record human motions. With a set of sensors installed on the bodies of the actors, the 3D positions of these sensors can be recorded by the computer. While such measurement can be both accurate and in real time, the motion capture process requires specific space setting, hardware and software to enable the process, making it a tool suitable only for one-time installation in a labo-

ratory, and this goes against our need for a system compatible to outdoor scenarios. (An example of domain gap caused by different dataset can be seen in Figure 3.2)

Figure 3.2: Domain gap is one of the main reasons why models trained on one dataset in a controlled environment struggle to generalize to other scenarios. In the example here, the model can perform a very accurate 3D prediction on the trained dataset in which the left image is located, but much less accurate for the 3D prediction on the right in which the background or the poses are out of the training distribution, where the orientation at which the human is facing is even wrong.

Instead, recording accurate human pose coordinates in 2D is much simpler, as simple as manually clicking on the image to indicate which pixel corresponds to a specific body part. Although collecting such information in large quantities can be a lot of work, it does not have a qualitative limit like recording 3D coordinates. Indeed, there already exists multiple outdoor "in the wild" datasets of 2D human poses with coordinate annotations, with which our step 1 can be trained with a better simulation of the scenarios we need for our system.

On the other hand, the whole step 2, which is to transform the 2D human pose into 3D, only requires the calculations between the 2D coordinates and the 3D coordinates, which means that the background environment does not affect this step. Thus, by dividing the monocular 3D human pose estimation into two distinct steps, we successfully create a walk-around of the limit of environment from 3D datasets.

3.3.2 Simple actions vs Professional actions

In addition to the high variation of different environments that we face in different working conditions, there is also a great diversity of different actions that individuals perform during work. From the overall pose of standing, sitting or lying on the ground, to hand actions ranging from lifting, operating in front of the head or natu-

ral sagging, these actions can be diverse depending on the type of work. Although existing datasets may be in-the-wild, most of these works are based on the most frequent simple actions, and none of these datasets cover human poses from different professional jobs. We can say that using the closest existing poses in the dataset we might be able to simulate the working poses, we still need a strong generalization of our system to cover the different actions. To tackle this problem, we propose the **Synthetic training for 2D to 3D human pose lifting**, presented in chapter 4, aiming to generate unseen poses from the dataset which the human pose distribution are realistic.

3.3.3 Capability of skeleton model

The capacity of the skeleton model represents the amount of information we can derive from the skeleton we choose, which is related to the geometric calculation in step 3. However, common state-of-the-art methods have two disadvantages . First, they only use a limited number of joints which represent only the most critical joints in the human body. (see Figure 3.3). Although these joints are capable of calculating the angle like flexion and extension, they are not capable of calculating the angle of supination-pronation, making these skeleton designs less preferable in this thesis. The second disadvantage is that the state-of-the-art methods deal with the face and hands independently of body information. They learn to recognize the joints of the body, hands and face separately and then combine them into one, but we prefer a combined structure of all information to make calculation easier. So we need to find a suitable skeleton layout for our project.

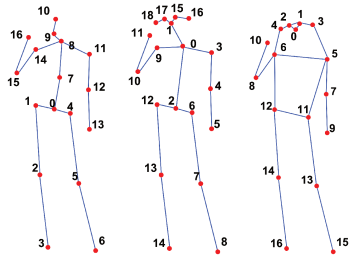

Figure 3.3: Several example of some most used skeleton structures, respectively from Human3.6m, CMU panoptic and COCO datasets. Image source [MMPose, 2020] MMPose, 2D/3D Body Keypoint Datasets

The first solution to this problem is a skeleton with a new arrangement of minimal joints that meets our need to calculate all angles. To achieve this, we design a skeleton with 32 joints shown in Figure 3.4 which the 32 red joints are to be estimated by the system, the 11 blue joints can be computed from red joints, and with all these joints we are able to calculate the angles we need. We called it **Ergonova skeleton**, named after the supporter of this thesis.

The advantage of this Ergonova skeleton is clearly its ability to calculate all the necessary angles we need, but it has a big disadvantage that no data is yet recorded

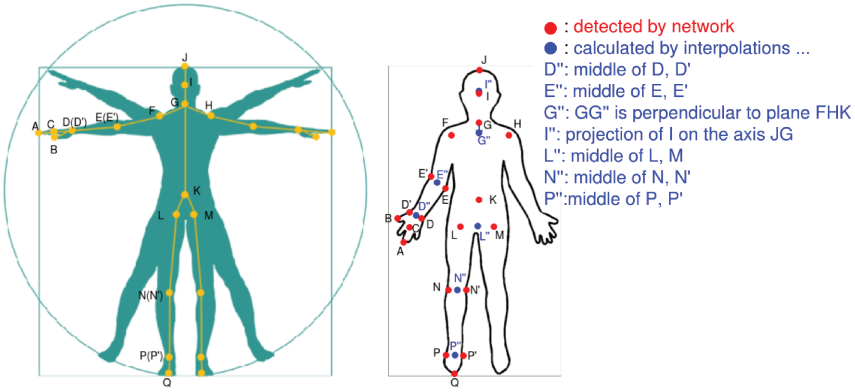

Figure 3.4: Our Ergonova skeleton capable of calculating all the angles we need. The skeletons on the left are the joints we need to consider to make the action recognizable, and with some duplication we obtain the skeleton on the right, containing 32 joints estimated from the system and 11 joints calculated with linear algebra from the previous joints, allowing the calculation of all rotation angles we need. Not all joints are shown in the image but can be seen with left-right symmetry.

under this skeleton, which makes the system incapable of being trained without first creating a dataset.

The second solution is to use an existing layout called **COCO-Wholebody** (see Figure 5.1) which is an extension of the COCO layout, but adding classic face, hands and feet information, making it a total of 133 joints, and there are already datasets using this layout as annotations, which means our network can be trained. However, it also has two weaknesses. One is that this layout is mainly annotated in 2D, while still lacking 3D dataset under this layout for research, another is that while this whole body skeleton layout greatly overlaps with our Ergonova skeleton, it lacks still some critical joints and it cannot directly calculate the supination and pronation angles of the arms and legs.

Our solution is to combine the strength of both skeletons. We choose to train with Wholebody skeletons, as well as create a 3D Wholebody dataset allowing training the network for 3D estimation, which is our contribution introduced in chapter 5. Then we derive the Ergonova skeleton from the whole body skeleton with interpolations of known joints, and finally calculate the angles from the Ergonova skeletons.

3.3.4 Real time demonstration

With the previous two works, we made a real-time demonstration that uses a computer camera to take an input image in real-time and perform 2D and 3D human pose estimation as well as critical (dangerous) pose detection. The detailed process and algorithm will be presented in chapter 6.

3.3.5 From pose to motion

While previous works are based on single image analysis, we would like to extent into motions where successive poses are temporally related. Unlike the most popular way of incorporating temporal information which is to use the sequence of images (video) as data to be fed into the system instead of a single image, we want to process motion sequences of which only a few frames are given, in order to recover missing intermediate poses, simulating recovery after data corruption when capturing data in real-world scenarios (For example, the case where the human motion inside the intermediate frames are completely occluded and unobserved, and we want to recover with the beginning and end frames where the human is still observable). With this motivation, we develop an algorithm that allows continuous interpretation of human poses over time with very limited input images, which is introduced in chapter 7.

Chapter 4

Synthetic training for 2D-3D human pose lifting

In this chapter we present our project of Synthetic Training for 2D-3D Human Pose Lifting, an algorithm which generates infinite synthetic 3D human skeletons on the fly during the training of the 2D-to-3D human pose lifter from just a few initial handcrafted poses. (see Figure 4.1).

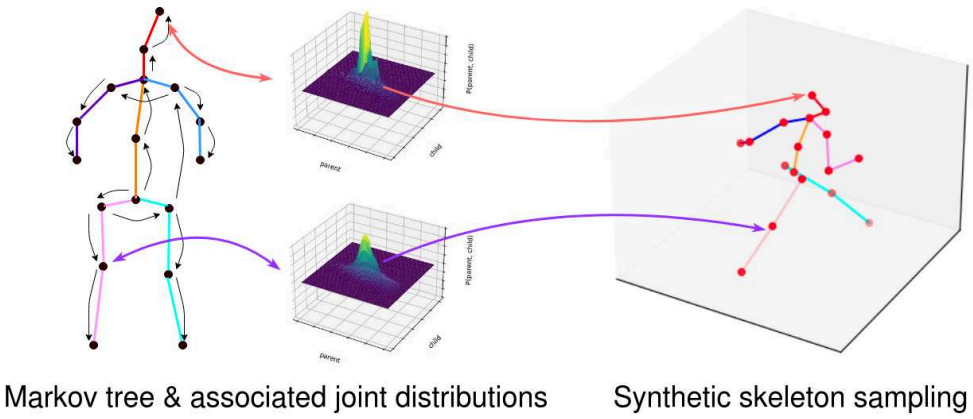

Figure 4.1: The main idea of our synthetic generation method: use a hierarchic probabilistic tree and its per joint distribution to generate realistic synthetic 3D human poses.

4.1 Synthetic human pose generation model

4.1.1 Basic skeleton model

Without loss of generalization, we use Human3.6M skeleton layout shown in Figure 4.3 (a) in this chapter, which covers the most useful joints of human body, including, the body vertebrae, arms, legs, and head orientations, allowing the visualisation of human poses and actions. This allows us to compare our results with the literature. For the reason of simplification, we set the pelvis joint (denoted joint 0) as root joint and the origin of the generation Cartesian coordinate system. After the generation of the poses, a non-zero value assigned to the root joint (as well as added to the other joints) is considered as the displacement of the whole human in the 3D coordinate space.

4.1.2 Local spherical coordinate system

From the nature of a human body, we suppose that the position of one joint depends on the position of the joint which is directly connected to it but closer to the root joint in geodesic meaning[1]. We call this kinematic chain **parent-child joint relations**, as shown in Figure 4.3 (b), from which a tree structure is applied to generate joints one by one. For each parent-child joint pair we propose to generate the child joint in a local spherical coordinate system (ρ, θ, ϕ) (see Figure 4.2) centered on its parent joint (see Figure 4.3 (d)).

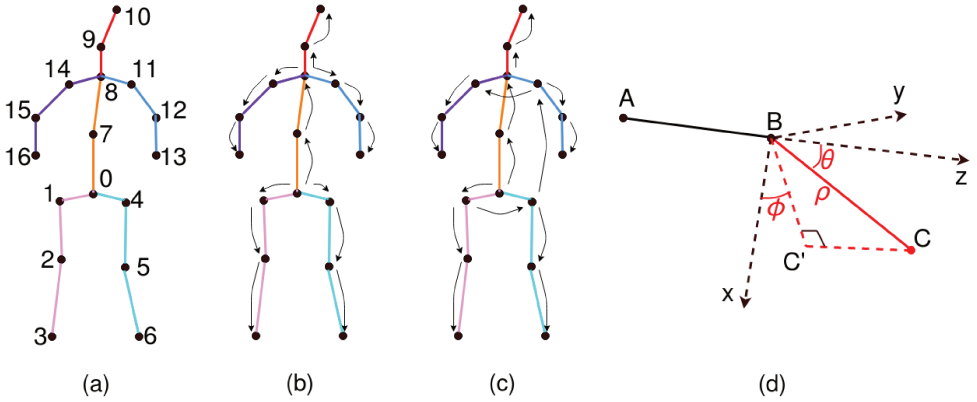

Figure 4.3: (a) The 17-joint model of Human3.6M that we use (b) The **parent-child joint relation** graph. With parent joint's coordinate as origin of local spherical coordinate system, it generates child joint's position. (c) The **parent-child ρ, θ , ϕ relation** graph. With parent joint's ρ, θ , ϕ information, it samples child joint's ρ, θ , ϕ. (d) An example of how child joint C is generated with sampled ρ, θ , ϕ from relationship in (c) under the local spherical coordinate system with it's parent joint B in (b) as origin and with axis dynamically depending on its parent branch $\vec{AB}$ orientation.

Our motivation to use such a local spherical coordinate system for joint generation is that each human body branch has a fixed length ρ no matter the movement. Also, since the supination and the pronation of the branches are not encoded in skeleton representation, the new joint position can be parameterized with polar angle θ and azimuthal angle ϕ. Furthermore, by using an axis system depending on 'grandparent-parent' branch instead of global coordinate system, the possible angle intervals of θ and ϕ achieved by human are more limited than in a global coordinate system, thus facilitate the generation process, and by controlling these angle intervals, it is more likely to generate realistic poses. Finally, our local spherical coordinate sys-

[1] Like left foot's position depends on position of left knee

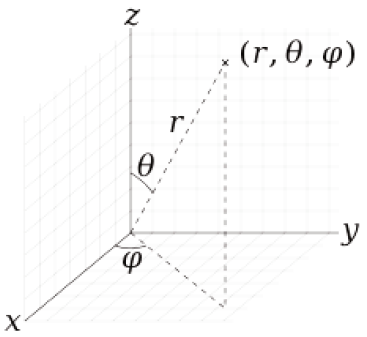

Figure 4.2: (r, θ, ϕ): the radial distance of the radial line r (annotated as ρ in our case) connecting the point to the fixed point of origin, the polar angle θ and the azimuthal angle ϕ. Image source: [wikipedia, 2001] Spherical coordinate system

tem is entirely bijective with global Cartesian coordinate system, allowing the simple transformation between generation spherical coordinate space and practical Cartesian coordinate space.

One point worth noting is that the shown axis in Figure 4.3 (d) in the generation space are defined based on the direction of the parent branch, meaning that 3 axis shown in the image here are not exact x,y,z-axis in the global Cartesian space which are horizontal or vertical. This setting allows that the biological achievable interval of ρ and θ of the children will not change no matter how its parent branch rotate.

4.1.3 Hierarchic probabilistic skeleton sampling model

Generating a human pose in our local spherical coordinate system is equivalent to generating a set of (ρ,θ,ϕ). We thus propose to sample these values from a distribution that approximate that of real human poses. To retain plausible poses, we limit the range of (ρ,θ,ϕ) for each joint based on what is on average biologically achievable.

Since body joints follow a tree-like structure, even though the biological achievable interval of ρ and θ of the children will not change no matter how its parent branch rotate, the probability of each angle inside the interval of the child $P(x_{child})$ can not still remains the same when the parent joint value changes, which the latter normally indicates a different action. It is unlikely that sampling each joint independently of the others leads to realistic poses. Instead, we propose to sample the ρ, θ , ϕ values with respect to a conditional distribution $P(x_{child}|x_{parent})$. This produces a Markov chain[2] indexed by a tree structure which we denote as a Markov Tree, More formally, denoting a child joint c and its parent $p(c)$ following the tree structure, we have:

$$(\rho_c, \theta_c, \phi_c) \sim P((\rho, \theta, \phi)|(\rho_{p(c)}, \theta_{p(c)}, \phi_{p(c)})) \tag{4.1}$$

Please note that the tree structure used for accounting the dependencies between joints as shown on Figure 4.3 (c) is slightly different than the kinematic one. We found in practice that it is better to condition the position of one shoulder on the position of the same side hip, and to condition symmetrical shoulder/hip on their already generated counterpart rather than on their common parent. Intuitively, this seems to

[2] [wikipedia, 2002]

A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event

better encode global consistency.

To facilitate modeling distribution $P((\rho,\theta,\phi)|(\rho_{p(c)},\theta_{p(c)},\phi_{p(c)}))$, we make further assumption that all 3 components only depend on their parent counterparts, or formally:

$$\rho_c \sim P(\rho|\rho_{p(c)}),\ \theta_c \sim P(\theta|\theta_{p(c)}),\ \phi_c \sim P(\phi|\phi_{p(c)}) \tag{4.2}$$

This allows us to model each distribution with a simple non-parametric model consisting of a simple 2D histogram representing the probability of sampling[3], In practice, we quantize each value into 50 bins histograms, totalling to $3 \times 16 = 48$ 2D histograms of size 50×50. When there is no ambiguity, we use the same notation $P(\cdot|\cdot)$ for the histogram and the probability.

[3] For example, $\theta_{left\ foot}$ knowing the value of $\theta_{left\ knee}$

4.2 Pseudo-realistic 3D human pose sampling

Once we set up a Markov chain model to allow the generation of the skeleton, next step is to estimate a distribution that can approximate the real 3D pose distribution, and from which our model can sample, so that the generated poses look like real human actions. Under the constraint of zero-shot 3D real data and purely 3D-synthetic, we choose to make breakthrough by looking at limited amount of 2D real poses and 'manually' lift them into 3D to make our distribution. However, it is impossible for us to tell the exact depths of keypoints from an image with our eye, and it is also a huge amount of work to do if we check a lot of images one by one. Instead, we choose a 3-step procedure to get our handcrafted 3D pose, which are:

1. Choosing high-variance 2D poses as seeds
2. Semi-automatic 2D to 3D seed pose lifting
3. Distribution diffusion

4.2.1 Choosing high-variance 2D poses as seeds

This step is to choose a few 2D samples with high variance allowing it to represent as many as different poses as possible. We randomly sample 1000 sets, each of which contains 10 different 2D-human poses from a real dataset[4]. We then compute the total variance for each set and pick the sets with largest variance as our candidates. This ensure our initial pose set has high diversity.

[4] For example, Human3.6M

4.2.2 Semi-automatic 2D to 3D seed pose lifting

This step is to achieve quasi-accurate 3D human poses as 3D seeds from previous 2D seeds. Since previous steps picked out only 10 poses in each set as seeds, manual-helped lifting work becomes doable. Here, we use a semi-automatic way to lift 2D seeds to 3D.

The idea is as follows: from an image for which we already know the 2D distances between connected joints, and if we can estimate the 3D length of each branch who connects the joints as well as the proportion λ_{prop} between the 2D length in the image (in pixel) and the 3D length (in centimeter), which in general can be considered as an equivalence of a camera's focal length, we can estimate the relative depth between connected joints using Pythagorean theorems under the assumption that the camera produces an almost orthogonal projection. The ambiguity about the sign of these depths, which decide if one joint is in front of or in the back of its parent joint, can easily be manually annotated.

To estimate the 3D length, we define a set of fixed value representing branch lengths ($||c - p(c)||_2, \forall c$ except the root joint)[5] of the human body based on biological data. While one may argue that different human individual should have different branch lengths of their bodies, since we later calculate under a proportionality assumption between 3D and 2D, we only need it to roughly represent the proportionality between different human bone length. We also manually annotate $sign_c$ for each keypoint c, denoting if it is relatively further or closer to the camera compared to its parent joint $p(c)$.

[5] For example, c=left foot means this is the branch between left foot and left knee, thus the left calf

The 2D-3D size proportion λ_{prop} is calculated under the assumption that the 3

joints around the head (head top, nose and neck as C, B, A respectively) form a triangle of known ratio which is independent of rotation and view, visually shown in Figure 4.4. This is reasonable since there are no largely moving articulated part in this triplet. We choose $AB = 1$ the unit length and we suppose the proportion between AB, BC and CA is fixed ($BC = \alpha AB$,$AC = \beta AB$). Noting $d_B = B'B - A'A$ and $d_C = C'C - A'A$, which can be positive or negative values, with A', B', C' the projection of A, B, C onto 2D camera plan respectively.

Now, for the 2D skeleton we already know $A'B'$,$B'C'$ and $A'C'$, then we have 3 unknown variables d_B, d_C, and $\lambda_{prop} = \frac{A'B'(pixels)}{A'B'(meters)}$ and 3 equations:

$$\begin{aligned} d_B^2 &= AB^2 - (\frac{A'B'}{\lambda_{prop}})^2, \\ d_C^2 &= (\beta AB)^2 - (\frac{A'C'}{\lambda_{prop}})^2, \\ (d_B - d_C)^2 &= (\alpha AB)^2 - (\frac{B'C'}{\lambda_{prop}})^2 \end{aligned} \tag{4.3}$$

Then we can solve λ_{prop}. In practice, we set $\alpha = 1$ and $\beta = 5/3$, which is a rough estimation from the author's head.

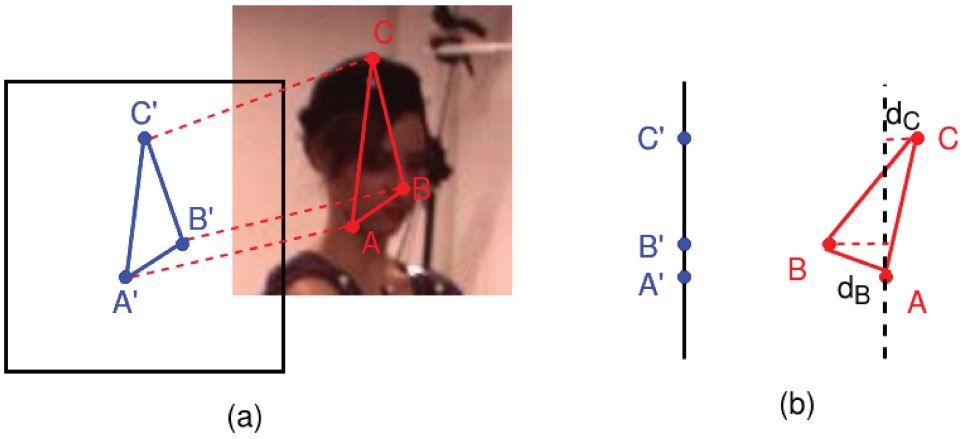

Figure 4.4: **(a)** 3D poses (red A,B and C, unit in centimeters) of 3 joints of the head projected onto 2D camera plan (blue A',B' and C', unit in pixels). **(b)** same but right side view after 90^o rotation.

After obtaining these depths, we apply Pythagorean theorem to get the final depth value of all joints with the kinematic order same as Figure 4.3 (b). Examples of semi-automatic lifted 3D poses are shown on Figure 4.5. Since there are only a few keypoints to label with *in front of* or *behind* their parent joint, the labeling process is

very easy and takes about 3 minutes per image only.

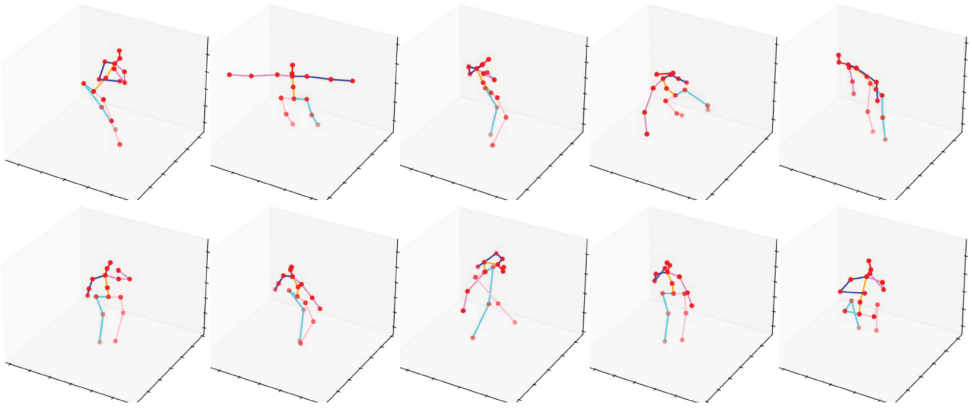

Figure 4.5: An example of a set of 10 semi-automatic lifted 3D poses. This set of seeds is also the one which produce our best score on Human3.6M dataset. These 10 lifted samples have a 79.42mm MPJPE error compare to the groundtruth.

4.2.3 Distribution diffusion

Since only 10 poses are not enough to estimate the real pose distributions, the goal of this step is to increase the variety of the poses from generation based on the initial seeds.

We transform 3D seeds into the local spherical coordinate system and used each seed set as initial distribution to populate the histograms. Since the sampling of a new skeleton follows the Markov tree structure and different limbs have a weak correlation between them in our model, it is possible to sample skeletons that look like combinations of the original 10 samples within the seed set.

However, these initial samplings are by no mean complete, and we run the risk of overfitting the lifter network to the combination of these poses only. To alleviate this problem, we introduce a diffusion process among each 2D histogram such that the probability of adjacent parameters is raised over time. More formally:

$$P(x_c|x_{p(c)})_{t+1} = P(x_c|x_{p(c)})_t + \alpha_{x_c}\Delta P(x_c|x_{p(c)})_t,\ x \in \{\rho,\theta,\phi\} \tag{4.4}$$

where Δ is the Laplacian operator and α_{x_c} is the diffusion coefficient. This idea is derived from the heat equation[6] in thermodynamics, in which bins with a higher prob-

[6] [wikipedia, 2003a]

In mathematics, if given an open subset U of R_n and a sub-interval I of R, one says that a function $u : U \times I \to R$ is a solution of the heat equation if $\frac{\partial u}{\partial t} = \Delta u$

ability diffuse to their neighbours (with Laplacian operator), making the generation process more and more likely to generate samples out of initial bin.

The main reason behind our diffusion process is that of curriculum learning. At first, the diversity of sampled skeletons is low and the neural network is able to quickly learn how to lift these poses. At later stage, the diffusion process allows the sampling process to generate more diverse skeletons that are progressive extensions of the initial pose angles, avoiding overfitting the original poses. We show in Figure 4.6 an example of evolution of the histogram and increase of generation variety through diffusion.

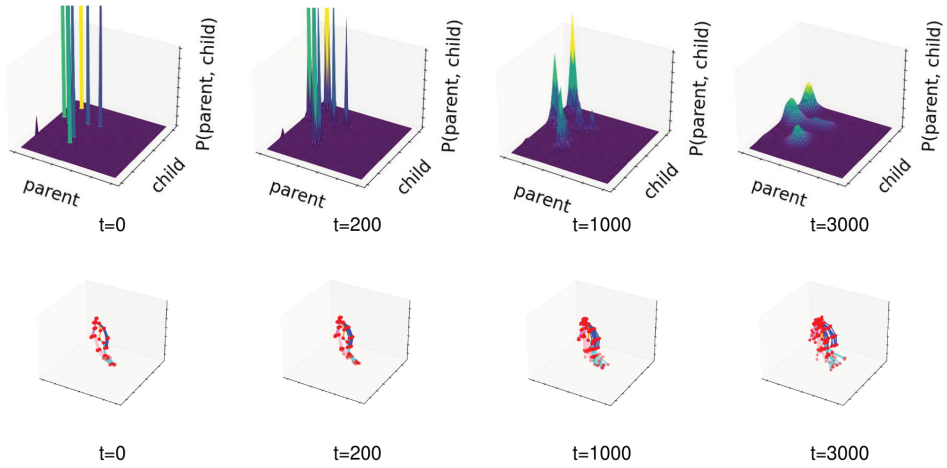

Figure 4.6: First row is an example of the distribution histogram of a joint after 0, 200, 1000 and 3000 steps of diffusion. Second row shows an example of slightly increased generation variety when sampling from a single bin and generating 10 samples each time after 0, 200, 1000 and 3000 steps of diffusion.

4.3 Training with synthetic data

The training setup of 2D-3D lifter network l_w is shown on Figure 4.7 and consists of 3 main components: (1) Sampling a batch of skeletons at each step ; (2) sampling different virtual cameras to project the generated skeletons into 2D ; and finally (3) the different losses used to optimize the lifter l_w. In practice, l_w is a simple 8-layer MLP with 1 in-layer, 3 basic residual blocks of width 1024, and 1 out-layer, adapted from Canonpose[7], shown in Figure 4.8

When sampling a new batch of skeleton using our generator, we have to keep in

[7] [Wandt et al., 2021]
Canonpose: Self-supervised monocular 3d human pose estimation in the wild

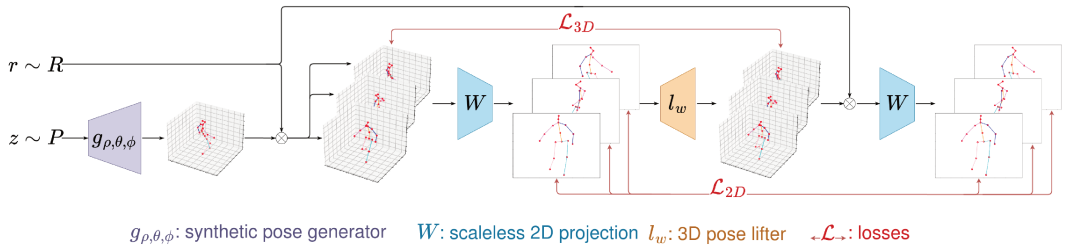

Figure 4.7: Our whole training process with synthetic data. Our generator g generates a 3D human pose following given distributions P of ρ, θ and ϕ. It will be applied with multiple different random generated r to project into different camera view. Projector W will projects them into scaleless 2D coordinates and they are the network inputs. The output estimated 3D poses will be applied with scaleless 3D supervision loss $\mathscr{L}_{3D}$, and also cross-view scaleless 2D reprojection loss $\mathscr{L}_{2D}$, which rotate estimated 3D pose from one view to another with known r and apply 2D supervision after projection W.

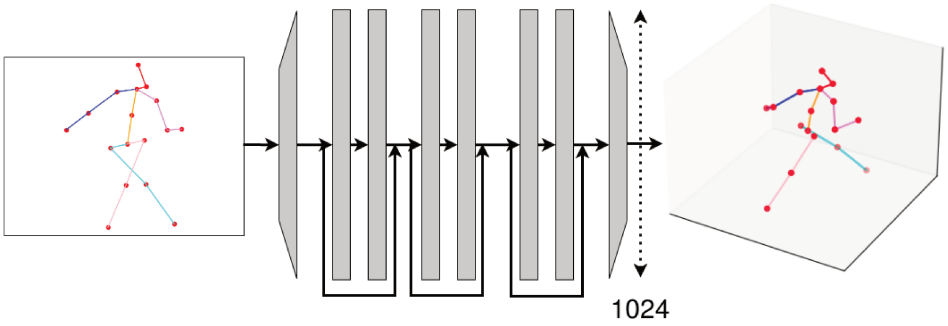

Figure 4.8: The simple 8 layer network for lifting task

mind that the distribution of the generator varies through time because of the diffusion process introduced in Equation 4.4. To avoid over-sampling or under-sampling bins with low density, we propose to track the amount of skeletons that have been generated in each bin and adjust the sampling strategy accordingly.

More formally, let us denote P_t the *true distribution* obtained by Equation 4.4, and P_e the *empirical distribution* obtained by tracking the generation process. The corrected sampling algorithm is shown in Algorithm 1 and basically selects uniformly a plausible bin ($P_t > 0$) that has not been over-sampled ($P_e \leq P_t$ means over-sampled). The whole generation process simply loops over all joints using the Markov tree and is shown on Algorithm 2.

As initialization of P_t, we sample 5000 real 2D poses, compute the proportion of nearest neighbour within each pose seed, and use it to initialize the histogram to give more importance to more frequent poses.

Algorithm 1 Sampling algorithm

Require: True distribution P_t, empirical distribution P_e;
  $bins \leftarrow$ where $P_t > 0$ and $P_e \leq P_t$
  $b \sim \mathscr{U}(bins)$
  return Random sample from b

Algorithm 2 Pose generation algorithm

Require: True distribution P_t, empirical distribution P_e, Markov tree structure T, sampling algorithm S
  $X \leftarrow 0_{(J,3)}$ ▷ 3=ρ, θ, ϕ
  for $i \in \rho, \theta, \phi$ **do** ▷ root joint
    $X[0,i] \leftarrow$ S($P_t(X_0), P_e(X_0)$)
  end for
  for (p,c) in T **do** ▷ parent-child relations in T
    for $i \in \rho, \theta, \phi$ **do**
      $X[c,i] \leftarrow \mathrm{S}(P_t(X_{(c,i)}|X_{(p,i)}), P_e(X_{(c,i)}|X_{(p,i)}))$
      Update $P_e(X_{(c,i)}|X_{(p,i)})$
    end for
  end for
  return X in Cartesian coordinates

Regarding the projection of the batch into 2D, we propose to sample a set of batch-wise rotation matrices $R_{1,\dots,N}$, mostly rotating around the vertical axis, to simulate different viewpoints. Then, the rotated 3D skeletons are just simply: $X_{3D,i} = R_i X_{3D,0}, \quad i \in \{1,\dots,N\}$, with $X_{3D,0}$ being the original skeleton in global Cartesian coordinates. To simulate the cameras, we use a scaleless orthogonal projection:

$$X_{2D,i} = \frac{W X_{3D,i}}{\|W X_{3D,i}\|_F}, \quad W = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \tag{4.5}$$

where W is the orthogonal projection matrix and $\|\cdot\|_F$ is the Frobenius norm. Normalizing by the Frobenius norm allows us to be independent of the global scale of $X_{2D,i}$ while retaining the relative scale of each bone with respect to each other. In practice, we found that uniformly sampling random rotation matrices at each batch renders the training much more difficult. Instead, we sample views with a small noise around the identity matrix and let the noise increase as the training goes on to generate more complex views at later stages.

Finally, to train the network, we leverage several losses. First, since we have the

3D ground-truth associated with each generated skeleton:

$$\mathscr{L}_{3D} = \frac{1}{N}\sum_{i=1..N}\left\| \frac{\hat{X}_{3D,i}}{\|\hat{X}_{3D,i}\|_F} - \frac{X_{3D,i}}{\|X_{3D,i}\|_F} \right\|_1, \tag{4.6}$$

with $\hat{X}_{3D,i} = l_w(X_{2D,i})$ being the output of the lifter l_w, and $\|\cdot\|_1$ the ℓ_1 norm. 3D skeletons are normalized before being compared because the input of the lifter is scaleless and as such it would make no sense to expect the lifter to recover the global scale of X_{3D}. Then, we use the multiple views generated thanks to R_i to enforce a multiview consistency loss. Calling $\hat{X}_{2D,i,j} = WR_jR_i^{-1}\hat{X}_{3D,i}$ the projection of the lifted skeleton from view i into view j, we optimize the cross-view projection error:

$$\mathscr{L}_{2D} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left\| \frac{\hat{X}_{2D,i,j}}{\|\hat{X}_{2D,i,j}\|_F} - \frac{X_{2D,j}}{\|X_{2D,j}\|_F} \right\|_1 \tag{4.7}$$

The global synthetic training loss we use is the following combination:

$$\mathscr{L} = \mathscr{L}_{2D} + \lambda_{3D}\mathscr{L}_{3D} \tag{4.8}$$

4.4 Experiments

4.4.1 Datasets

We use two widely used dataset Human3.6M[8] and MPI-INF-3DHP[9] to quantitatively evaluate our method.

We only use our generated synthetic samples for training and evaluate on S9 and S11 of Human3.6M and TS1-TS6 on MPI-INF-3DHP with their common protocols. In order to compare the quality of our generated skeletons with real 2D data, We also use the COCO[10] and MPII[11] datasets to check the generalizability of our method with qualitative evaluation.

4.4.2 Evaluation metrics

For the quantitative evaluation on both Human3.6M and MPI-INF-3DHP we use MPJPE, i.e. the mean euclidean distance between the reconstructed and ground-truth 3D pose coordinates after the root joint is aligned ($P1$ evaluation protocol of

[8] [Ionescu et al., 2014a]

Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments

[9] [Mehta et al., 2017]

Monocular 3d human pose estimation in the wild using improved CNN supervision

[10] [Lin et al., 2014]

Microsoft coco: Common objects in context

[11] [Andriluka et al., 2014]

2d human pose estimation: New benchmark and state of the art analysis

Human3.6M dataset). Since we train the network with a scaleless loss, we follow Canonpose[12] and scale the output 3D pose's Forbenius norm into the ground-truth 3D pose's Forbenius norm in order to compute the MPJPE. We also report PCK, i.e. the percentage of keypoints with the distance between predicted 3D pose and ground-truth 3D pose is less or equal to half of the head's length.

[12] [Wandt et al., 2021]

Canonpose: Self-supervised monocular 3d human pose estimation in the wild

4.4.3 Implementation details

We use a batch-size of 32 and we train for 10 epochs on a single 16G GPU using Adam optimizer and a learning rate of 10^{-4}. We set the number of views $N = 4$ and the total number of synthetic 2D input samples for each epoch is the same as the number of H36M training samples to make a fair comparison. The distribution diffusion coefficient α_{x_c} is a joint-wise loss dependent value, set to $10^{-5} \times 10^{|\delta\mathscr{L}|/(10\times N)}$ where $\delta\mathscr{L}$ is the joint-wise difference between loss of the last batch and the current batch, and the rotation R are sampled with a noise that increases in $\frac{1}{2\times \#batch}$ after each step, with $\#batch$ the number of elapsed batches in the current epoch. For the loss, $\lambda_{3D} = 0.1$ is set empirically. To account for the variation due to the selection of the 2D pose using total variance, we keep the 10 sets with highest variance and show averaged results. Our method trains on about 100k generated samples per hour on a V100 GPU, whereas inference time for lifting is negligible.

4.4.4 Comparison with the state-of-the art

We compare our results with the state-of-the-art methods with synthetic supervision for training in Table 4.1. We present several weak supervision methods which also do not use real 3D annotations, and instead use other sort of real data supervision whereas we do not. We can see that our method outperforms these synthetic training methods and achieves the performance on par with weakly supervised methods on H36M, while never using a real example for training.

We show qualitative results on the COCO dataset on Figure 4.9. It worth notice that the COCO layout is different from that of H36M, we use a linear interpolation of existing joints to localize the missing joints. We can see that our model still achieves good qualitative performances on zero shot lifting of human poses in the wild (first

	H36M	3DHP	
	MPJPE↓	MPJPE↓	PCK↑
Weak supervision			
[Iqbal et al., 2020]	67.4	109.3	79.5
[Mitra et al., 2020]	120.95	-	-
[Wandt et al., 2021]	65.9	104.0	77.0
Synthetic training			
[Li et al., 2020]	106.8	-	-
[Ghezelghieh et al., 2016]	$\geq$ 78.13	-	-
[Du et al., 2016]	126.47	-	-
[Varol et al., 2017]	111.6	-	-
Ours			
10 sets	95.4±13.5	148.4±7.6	57.7±2.3
best run	60.8	132.8	61.9

Table 4.1: Comparison of our results with the state-of-the-arts under the common protocol 1 on Human 3.6M and MPI-INF-3DHP. The value before and after ± symbol are mean and standard deviation values.

2 rows). Failed predictions (last row) tend to bend the legs backward even when the human is standing still, which may be a bias of the generator.

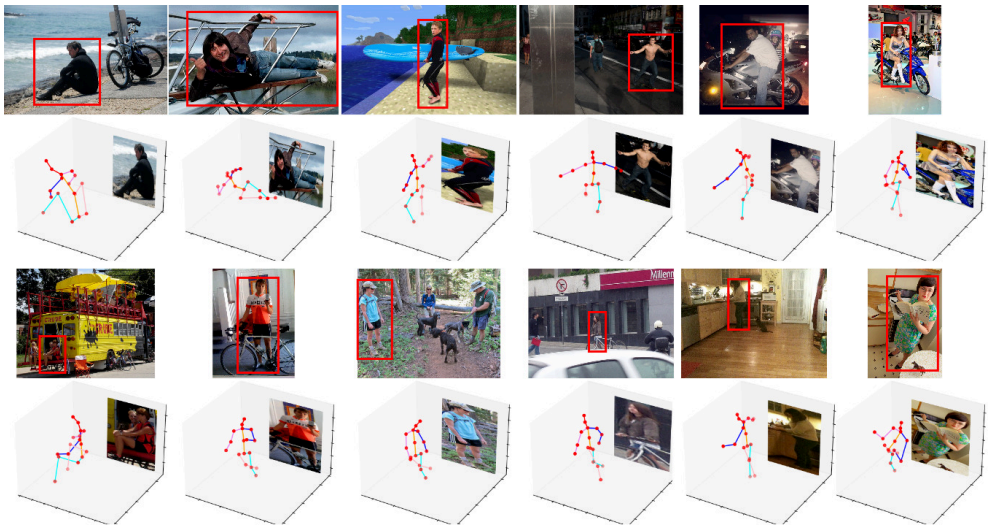

Figure 4.9: Example of zero shot lifting in the wild on images from the COCO dataset. The first row are visually correct prediction, while the last row presents 'failure' cases, mostly due to right leg learnt a bias of leaning backward.

4.5 Details studies

4.5.1 Synthetic poses realism

We want to see how similar our synthetic skeletons are to real skeletons. Qualitatively we compare our distribution after diffusion with the distribution of the whole

Human3.6M and MPI-INF-3DHP datasets, for some of the joints as shown in Figure 4.10. We can see that, although many poses in MPI-INF-3DHP never appear in Human3.6M, the angle distributions θ and ϕ of these two real datasets have very similar shapes, meaning our local spherical coordinate system successfully models the invariance of biologically realizable human pose angles and frequencies that are independent of camera viewpoint. Our seeds + diffuse strategy produces a Gaussian mixture which succeed in covering big parts of real dataset's distribution.

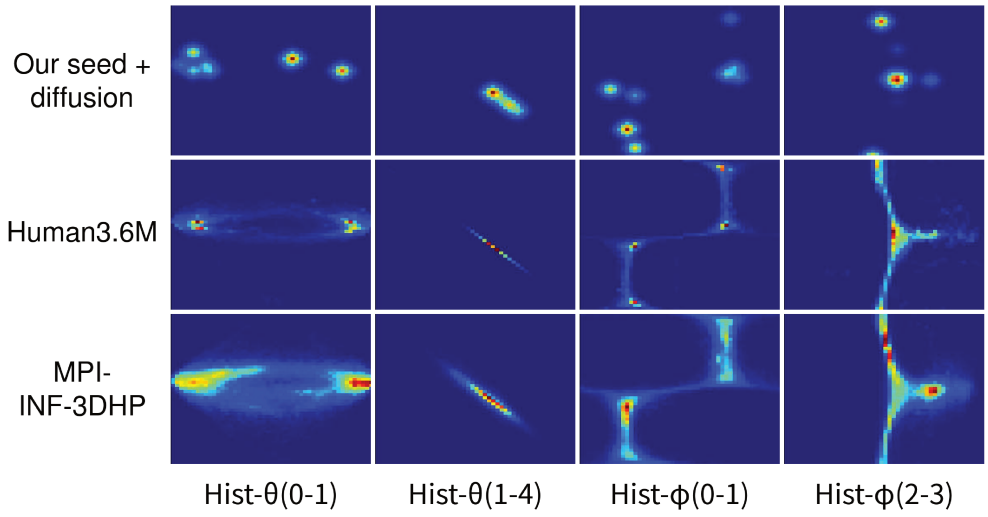

Figure 4.10: Examples of distributions of angle θ and ϕ from same parent-child pairs computed on Human3.6M, MPI-INF-3DHP, and our diffusion process.

Quantitatively we apply a precision/recall test, as is common practice with GANs[13]. We sample 5000 real and 5000 synthetic poses and project them to 2D plane using the scaleless projection in Equation 4.5 and the Euclidean distance. Precision (resp. Recall) is defined as percentage of synthetic samples (resp. real samples) inside the union of the balls centered on each real sample (resp. synthetic sample) and with a radius of the distance to its 10-th nearest real sample neighbor (resp. synthetic sample neighbor). In our case, we already know that most synthetic skeleton generated by our Markov tree are biologically possible thanks to the limits in the generation intervals. As such, we are more interested in a very high recall so as to not miss the diversity of real skeletons. All our seed sets have more than 70% recall and highest one achieves 91.8% recall. The precision, on the other hand, is around 40%, with 47.1% as the highest, which is still good considering we only start with 10 manually

[13] [Naeem et al., 2020]
Reliable fidelity and diversity metrics for generative models

lifted initial poses for each seed.

4.5.2 Effect of diffusion

We want to see why the diffusion process is essential to our method. We take respectively 1, 10, 100, 1000 and 10000 samples of 3D poses on the Human3.6M dataset as an initial seed to create distribution graphs and apply our 2D precision recall test after the diffusion process. The result is shown in Figure 4.11. We can see that dif-

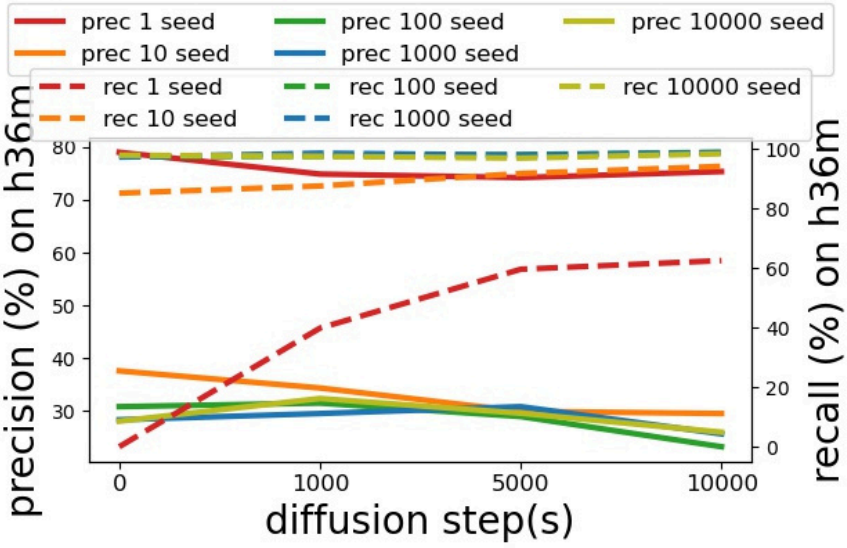

Figure 4.11: Precision and recall evaluated with 5k generated samples and 5k real 2D samples from h36m.

fusion generally increase recall value at the cost of precision value. The distribution using 1 samples as seed is much worse with the others in recall, meaning it can only cover around 60% of samples from a real dataset even with diffusion process, while the distribution using 100 or more samples are close in performances. The diffusion process can reduce the gap between the distribution using 10 samples as seeds and those using 100 or more samples, which is important to us considering we want to avoid handcrafting a lot of initial poses.

4.5.3 Layout adaptation

We show that our synthetic generation and training method also work on a different keypoint layout by applying the whole process on a newly defined hierarchic Markov

tree based on 24 keypoints of SMPL model[14] and evaluating on 3DPW dataset[15]. We use 24 samples from its training set (one frame from each video) using our 2D variance based criterion for seeds. Since our training method is scaleless, we rescale the predicted 3D poses by the average Forbenius norm of the 24 samples in the seed. The average MPJPE of 10 different seeds is shown in Table 4.2. The close results validates the generalization capability of our method.

[14] [Loper et al., 2015a]

SMPL: A skinned multi-person linear model

[15] [von Marcard et al., 2018b]

Recovering accurate 3d human pose in the wild using imus and a moving camera

Method	Labeled training data	MPJPE↓
[Li et al., 2022]	H36m + 3DHP + COCO + MPII + 3DPW	**52.8**
[Guan et al., 2022]	H36m + 3DPW	65.5
ours	24 samples from 3DPW	61.09 ± 2.16

Table 4.2: Results on the 24-keypoint SMPL model, compared to the state-of-the-art

4.5.4 Semi-automatic lifting

We want to see if our 'semi-automatic' pose lifting can be replaced with a fully-automatic algorithm, where we do not need to manually check the depth of each joint relative to its parent joint. We design an algorithm, which decides whether a joint is further or closer to the camera than its parent joint, based on the orientation of the current 2D pose (facing or back to the camera), values of the parent joint and a random seed generated for each joint. We compare the following 4 possible different ways of getting 3D poses from same 2D sample set of our seed:

1. directly take the correspondent 3D groundtruth poses' value of the 2D pose
2. using a pretrained lifter network using method from Canonpose[16]
3. Our semi-automatic lifting algorithm
4. Our full automatic algorithm, with a conditional probability for a joint being forward or backward of its parent joint

[16] [Wandt et al., 2021]

Canonpose: Self-supervised monocular 3d human pose estimation in the wild

And the MPJPE distance between each pairs are shown in Table 4.3. It shows that if we want to avoid using 3D data from a real dataset, using a pretrained network can be a solution, but we can still argue that the pretrained network itself has learned the prior from the real data. Oh the other hand, manual part of deciding forward/backward of a joint is important for us to make our seeds without any 3D real

compare methods	MPJPE(mm)
(1) & (2)	50.66
(1) & (3)	93.95
(1) & (4)	236.69
(2) & (3)	111.39
(3) & (4)	236.23

Table 4.3: Comparison of the MPJPE error between 4 different method of lifting 2D to 3D. Every estimated shape are normalized into same forbenius norm as the ground-truth 3D pose and root joint aligned.

data. The fully automatic algorithm, with several joints' forward/backward statues opposite to the ground-truth situation, can immediately produce much higher error. This necessity of the 'manual' part largely limit us of making seeds with a lot of samples.

4.5.5 Diffusion hyper-parameters choosing

To investigate the diffusion speed influence, we show the MPJPE and PCK on H36M and 3DHP for different speeds varying from 10^{-3} to 10^{-6} on Figure 4.12. As can be seen, diffusing slowly improves both MPJPE (solid lines) and PCK (dashed lines) compared to static distribution, but diffusing too quickly degrades the performances. We believe this is because higher diffusion speed tends to produce uniform histograms that generate unrealistic poses that the lifter struggles to learn.

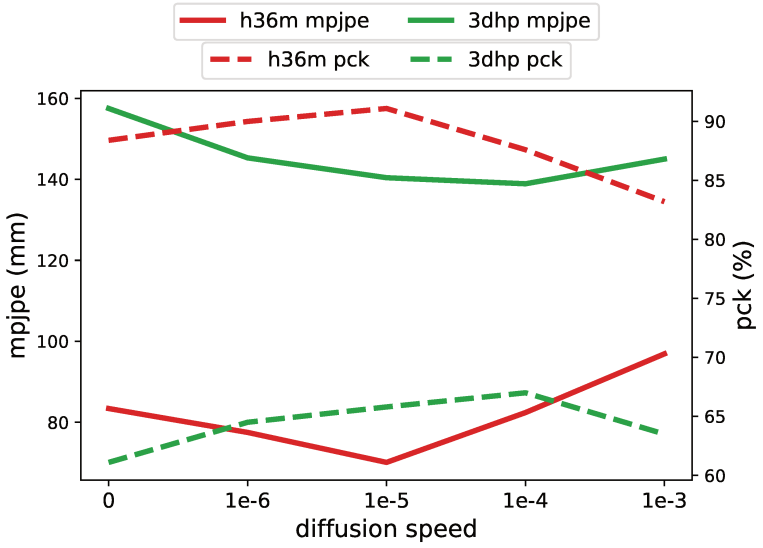

Figure 4.12: Comparing different diffusion speed with our handcrafted initial distribution.

We also investigate whether starting the diffusion process before training the lifter may improves the results. The idea is that otherwise, the network might overfit the first initial poses because of the lack of diversity and end up stuck in a bad local minimum, unable to further train for the full diversity of the diffused distribution, but according to Figure 4.13 and Table 4.4 the benefit does not seem to be significant.

Training variables			H36m eval		MPI-INF-3DHP	
strategy	size	pre-step	MPJPE	PCK	MPJPE	PCK
random H36M 3D	100	0	105.1	77.0	174.7	55.5
		1000	106.0	76.1	180.4	53.6
		10000	115.2	74.0	181.4	53.1
random H36M 3D	10	0	121.6	71.9	191.3	51.7
		1000	124.1	70.9	192.9	51.1
		10000	126.6	67.9	194.5	50.4
random H36M 2D	100	0	91.6	84.1	128.1	70.9
		1000	91.4	84.8	128.5	71.0
		10000	91.5	84.3	128.0	71.2
random H36M 2D	10	0	103.0	78.7	141.5	66.9
		1000	103.9	79.3	141.6	67.0
		10000	105.6	78.4	140.7	67.3
fixed H36M 2D	10	0	74.9	88.5	143.1	65.6
		1000	74.4	87.8	140.5	66.6
		10000	78.5	87.0	140.7	66.4
handcraft	10	0	80.0	89.6	144.2	65.2
		1000	70.1	91.1	140.4	65.8
		5000	78.4	88.8	142.4	65.8
		10000	81.8	89.6	141.6	67.0

Table 4.4: Comparing different pre-diffusion step with different initial distribution with diffusion speed of 10^{-5}.

The results of the two graphs show that, a diffusivity coefficient that is too large will bring the distribution towards uniform too fast and less specifically on poses close to the initial poses, thus will lower the performance on H36M but increase performance on 3DHP. A diffusion coefficient around 10^{-5} seems to be the most balanced. On the other hand, a pre-diffusion of 1000 appeared to be the best choice, which neither overfit the initial 10 poses, nor diffuse too much that the network can not learn well at the beginning.

4.6 Limitations

Even though we have achieved considerable results, there are still a few limitations of our method that we would like to address in the future:

- Our model is built based on Human3.6M 17 joint skeletons, which do not contain some information like eyes and ears. So in order to apply our method on

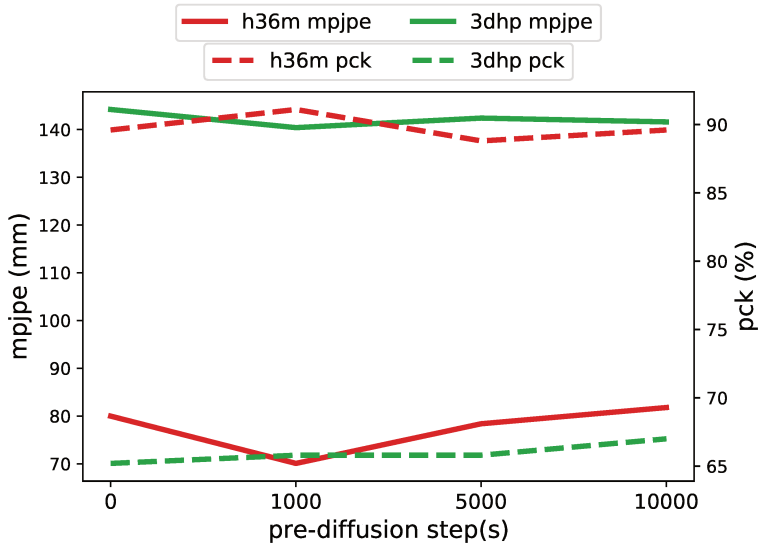

Figure 4.13: Comparing different pre-diffusion step 0, 1000, 5000, and 10000 with our handcrafted initial distribution with diffusion speed of 10^{-5}.

e

other skeletons like COCO, certain adaptations to the skeleton and the Markov tree of distribution, including both joints relations and ρ,θ,ϕ relations, are required differently for different skeleton models.

- In order to be generalizable across different scenarios, we adapt a scaleless training process, which means we can never obtained the exact size of the estimated human, unless we have some prior knowledge from other places.

- The choice of initial samples for seeds largely affects the training results. The seed that can give best evaluation result on the Human3.6M dataset is clearly an overfitting seed. To achieve greater generalibility, we need to use more diverse initial poses (like sleeping, upside down, etc.) to cover a greater distribution probability, but at the cost of lower performance on specific scenarios.

4.7 Conclusion

We present an algorithm that can generate synthetic 3D human skeletons on the fly during training, following a Markov-tree type distribution that evolve through out time to create unseen novel poses. We do not use any 3D values from existing 3D

datasets and only use a small limited amount for 2D human poses from real dataset to build up initial distribution. We use a K-NN based precision-recall evaluation metric to demonstrate how similar our generated skeleton is to real human poses. Based on the generated synthetic skeleton, we propose a scaleless multiview training process based on purely synthetic skeletal data generated from a few handcrafted poses. We evaluate our approach on the two benchmark datasets and achieve promising results in a zero-shot setup.

Chapter 5

Wholebody 3D estimation

In this chapter we present our project of H3WB: Human3.6M 3D WholeBody Dataset and Benchmark, including a methodology to get accurate 3D wholebody annotations from multi-view images with only algorithms and no other devices, as well as several datasets making exploring 3D wholebody possible for the community.

5.1 Wholebody model

In the past, keypoint based 3D whole-body pose estimation has not been fully explored in the literature due to the absence of a representative and accurate benchmark. Existing 3D whole-body methods either rely on specific datasets and models for different body parts, leading to complex training pipelines and heterogeneous evaluations, or utilize parametric models that prioritize shape capture over highly precise keypoints. In addition, unified methods vary significantly in terms of keypoint layout definition, number of keypoints and distribution of keypoints across body parts (see Table 5.1). These significant dataset disparities and the absence of a standard benchmark make it challenging to compare methods fairly.

However, the work of combining all body part into a whole has already exist in 2D, with COCO-Wholebody[1] provided a layout of 133 keypoints (see Figure 5.1) , consists of 17 body keypoints, 6 foot keypoints, 68 face keypoints and 42 hand keypoints. Also existing work Openpifpaf[2] already has well pretrained model and weights for 2D wholebody pose detection.

[1] [Jin et al., 2020]

Whole-Body Human Pose Estimation in the Wild

[2] [Kreiss et al., 2021]

OpenPifPaf:Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association

Dataset	Size	Keypoints	Body	Hand	Face
Human3.6M [Ionescu et al., 2014a]	3.6M	17	17		
3DPW [von Marcard et al., 2018b]	51k	24	24		
LSP [Johnson and Everingham, 2010]	10k	14	14		
3DHP [von Marcard et al., 2018b]	>1.3M	17	17		
Panoptic [Joo et al., 2015]	1.5M	15	15		
MTC [Xiang et al., 2019]	834K	20	20		
InterHand2.6M [Moon et al., 2020]	2.6M	21		21	
FreiHAND [Zimmermann et al., 2019]	37k	21		21	
RHD [Zimmermann and Brox, 2017]	44K	21		21	
MTC [Xiang et al., 2019]	111K	21		21	
TotalCapture [Joo et al., 2018]	1.9M	127	21	16+16	74
ExPose [Choutas et al., 2020]	33K	144	25	15+15	89
H3WB	100k	133	23	21+21	68

Table 5.1: Overview of datasets for 3D human pose estimation.

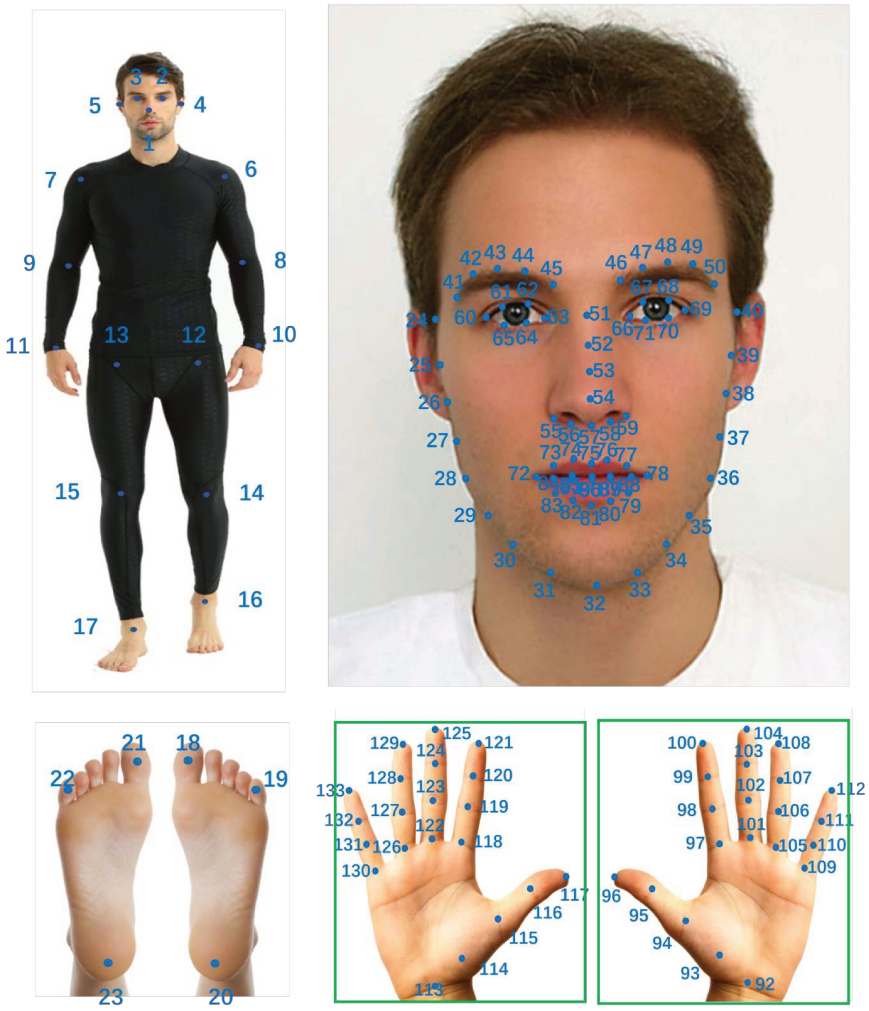

Figure 5.1: COCO-Wholebody layout, which consists of 17 for body, 6 for foot, 68 for face and 42 for hands, for a total of 133 keypoints. Image source: [Jin et al., 2020] Whole-Body Human Pose Estimation in the Wild

Based on this wholebody model, we build a few new datasets and a benchmark for keypoint-based 3D wholebody pose estimation. It is called Human3.6M 3D Whole-Body, or H3WB for short (see Figure 5.2), because our first method-testing dataset is an extension from training sequences of Human3.6M dataset[3] with 3D whole-body keypoint annotations. We later on also make 3D whole-body annotations for the training sequences of CMU Panoptic dataset[4] and MPI-INF-3DHP dataset[5] with the same data-construction algorithms but minor hyper parameter changes.

[3] [Ionescu et al., 2014a]

Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments

[4] [Lab, 2001]

Motion capture database

[5] [Mehta et al., 2017]

Monocular 3d human pose estimation in the wild using improved cnn supervision

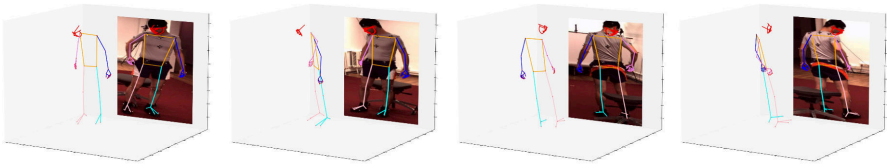

Figure 5.2: The H3WB dataset has 133 whole-body keypoint annotations in 3D as well as their respective projections in 2D.

5.2 The H3WB dataset

In short, the H3WB dataset building process is as follows: First, we use an off-the-shelf 2D whole-body detector combined with multi-view reconstruction to obtain an initial set of incomplete 3D whole-body keypoints. Next, we implement a completion network to fill in the keypoints missed by the multi-view geometric approach. Then, we develop a refinement method for the hands and the face to obtain more accurate keypoints. Finally, we perform quality assessment to select 3D whole-body poses with high confidence.

Without loss of generalization, we still use Human3.6M as our example dataset to present our 3D wholebody annotation making algorithm in this section.

5.2.1 Initial 3D whole-body dataset with OpenPifPaf

[6] [Kreiss et al., 2021]

OpenPifPaf:Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association

[7]

Subjects S1, S5, S6, S7 and S8, 1 image per 5 frames

We run the 2D whole-body detector from OpenPifPaf[6] on all 4 different camera views from the training set of Human3.6M[7]. Since the cameras of Human3.6M are well calibrated, we can reconstruct keypoints in 3D using a multi-view geometry.

The multi-view geometry algorithm does the following thing: For each adjacent pair of camera views (noting sub index 1 and 2), if a keypoint has been detected in both views, saying (u_1, v_1) and (u_2, v_2) respectively (shape 2×1), and we know the respective camera intrinsic matrices K_1 and K_2 of the views (shape 3×3), as well as relative rotation matrix R (shape 3×3) and translation matrix T (shape 3×1) from camera 2 to camera 1, we then have two depth scalar values d_1, d_2 and two 3d coordinate X_1, X_2 (shape 3×3) in two camera space, as well as three camera projection or transformation equations to solve them:

$$\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} d_1 = K_1 X_1$$
$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} d_2 = K_2 X_2 \tag{5.1}$$
$$X_1 = R X_2 + T$$

Placing first two equations, the 3D-to-2D camera projection matrices into the third equation, we have:

$$K_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} d_1 = R K_2^{-1} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} d_2 + T \tag{5.2}$$

Noting $A = R K_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$ and $B = K_2^{-1} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}$ which are both known variables of shape 3×1 (3 rows, 1 column matrix), same as T, the equation becomes:

$$A d_1 + T = B d_2 \tag{5.3}$$

Since we will never know if Equation 5.3 always have an exact solution for every data sample, we choose to find the value d_1, d_2 to minimize the norm $||A d_1 + T - B d_2||_2$, which leads to the new equations:

$$\begin{pmatrix} A^T A, -A^T B \\ A^T B, -B^T B \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = - \begin{pmatrix} A^T T \\ B^T T \end{pmatrix} \tag{5.4}$$

with A^T and B^T transpose matrices of A and B respectively, and all $A^T A$, $A^T B$, $B^T B$, $A^T T$ and $B^T T$ are 1×1 scalar. Its solution is:

$$d_1 = \frac{B^T B \times A^T T - A^T B \times B^T T}{A^T B \times A^T B - A^T A \times B^T B}$$
$$d_2 = \frac{A^T B \times A^T T - A^T A \times B^T T}{A^T B \times A^T B - A^T A \times B^T B} \tag{5.5}$$

Back projecting into Equation 5.2, we can obtain 3D coordinates X_1 and X_2, as well as computing the coordinates X under global coordinate system. After taking the average between all pair of camera views, we obtain the initial 3D whole-body annotations.

Still there are some drawbacks. The OpenPifPaf 2D whole-body detector can miss keypoints due to self-occlusions (hands, feet) or unfavorable camera viewpoints (face backward). However, the 4-view setup allows us to recover missing keypoints and obtain a complete 3D whole-body pose, provided each keypoint appears in at least two non-opposing views. An example of this process is shown in Figure 5.3. Using this method, we obtained 11,426 fully complete 3D whole-body poses with

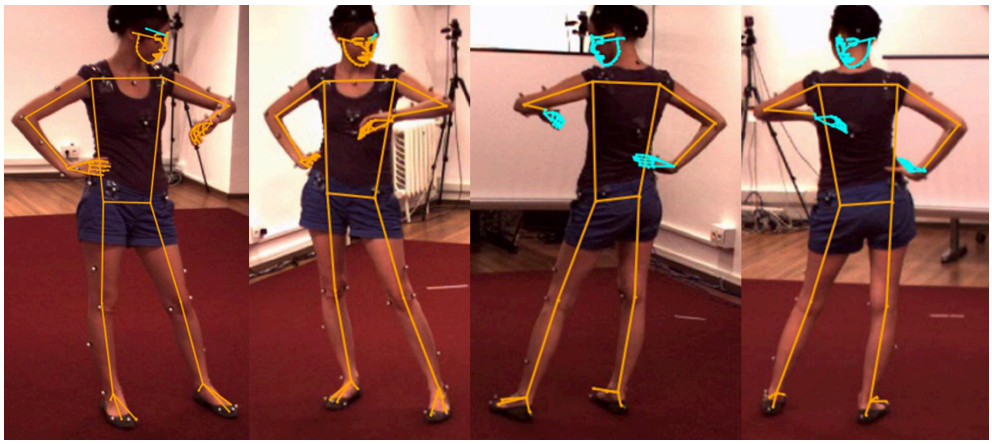

Figure 5.3: OpenPifPaf detects most of the non-occluded keypoints inside the image (orange keypoints). The occluded or undetected keypoints (cyan keypoints) are re-projections after 3D multi-view reconstruction. Notice that these reprojections do not always align with the images, like the right hand in the last view, which is probably due to OpenPifPaf not being perfectly accurate.

all 133 keypoints and 26,333 incomplete 3D whole-body poses where all keypoints appear in at least one view, resulting in a total of 37,759 3D whole-body poses with each keypoint appearing in at least one view.

5.2.2 Completion network

In order to complete the 26,333 incomplete 3D whole-body poses, we develop a completion network as shown in Figure 5.4. We design our completion network using Transformer architecture[8] as they can easily handle the conditional dependencies introduced by the skeleton's topology through masking. Since each skeleton always has exactly 133 keypoints, which can be considered as 133 tokens of 3 coordinate values. Token values are expanded from 3 coordinates to $3 \times 16 = 48$ features using Fourier encoding. We use learnable positional encoding since each keypoint is uniquely identified.

[8] [Vaswani et al., 2017] Attention is all you need

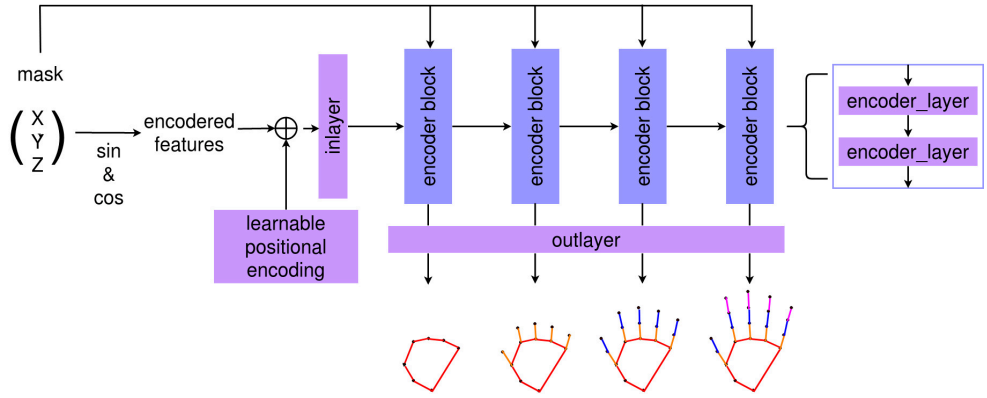

Figure 5.4: The completion network consists of one linear input layer, 4 transformer encoder blocks (each of them containing 2 transformer encoder layer with $d\_model = 64$ and $n\_head = 1$), and a linear output layer. At the end of each encoder block, the features are decoded by the output layer into a predicted position in a curriculum way where later blocks decode more keypoints.

We train the completion network on the 11,426 complete skeletons using a masked auto-encoder strategy[9] where the missing keypoints are masked at the input and will be predicted using the unmasked keypoints. The masking strategy is as follows:

[9] [He et al., 2022b] Masked autoencoders are scalable vision learners

- With a 50% chance, we perform a keypoint wise mask where each keypoint has 15% chance of being masked,
- with the remaining 50% chance, we perform a block wise mask in which either the body, the left hand, the right hand, the left or the right part of the face are masked (uniform probability).

To ease the learning process and take into account the causal link between some keypoints[10], we introduce a curriculum approach. We compute the loss at different

[10] For example, the tip of a finger depends on the position of its parent phalanges

levels following a hierarchy where early levels consider only keypoints closer to the root, while later levels consider more deformable keypoints which highly depend on their parents. We illustrate the completion network and learning process in Figure 5.4. The loss function is

$$\begin{aligned}\mathscr{L}(X, X_{gt3D}, X_{gt2D}) =& \mathscr{L}_{3D}(X, X_{gt3D}) \\ &+\alpha \mathscr{L}_{2D}(X, X_{gt2D}) \\ &+\beta \mathscr{L}_{sym}(X),\end{aligned} \tag{5.6}$$

where $\mathscr{L}_3D$ is an ℓ_1 loss of 3D coordinates, $\mathscr{L}_2D$ is an ℓ_1 loss of 2D projection of the 3D coordinates if we have the 2D annotation from OpenPifPaf, and $\mathscr{L}_{sym}$ is a symmetric loss which is applied to make sure the left part and right part of the human have the same length on corresponding body parts.

We show an example output from our completion network in Figure 5.5. The completion network results on missing body parts are visually realistic and appealing. However, since the completion network does not rely on the image content, its output does not always align with the image and may only reflect the most common poses of the training set.

5.2.3 Hands and face 2D refinements

In order to correct the alignment problem, we propose another neural network that refines the 2D position of keypoints on the face and the hands. Previous studies have explored and demonstrated the effectiveness of 2D human pose refinement using an iterative error feedback framework[11]. Motivated by this, we build upon recent conditional diffusion models[12] and we consider the prediction from the completion network as *noisy* such that the refinement network *denoises* it to conditionally fit the image.

We train separate refinement models for the face and the hands, while keeping the same network architecture and the same training strategy. We used a simple MLP and found it to be effective, preventing the need to explore more complex architectures. We illustrate the refinement process in Figure 5.6. During training, we add Gaussian

[11] [Carreira et al., 2016]

Human pose estimation with iterative error feedback

[12] [Ho et al., 2020]

Denoising diffusion probabilistic models

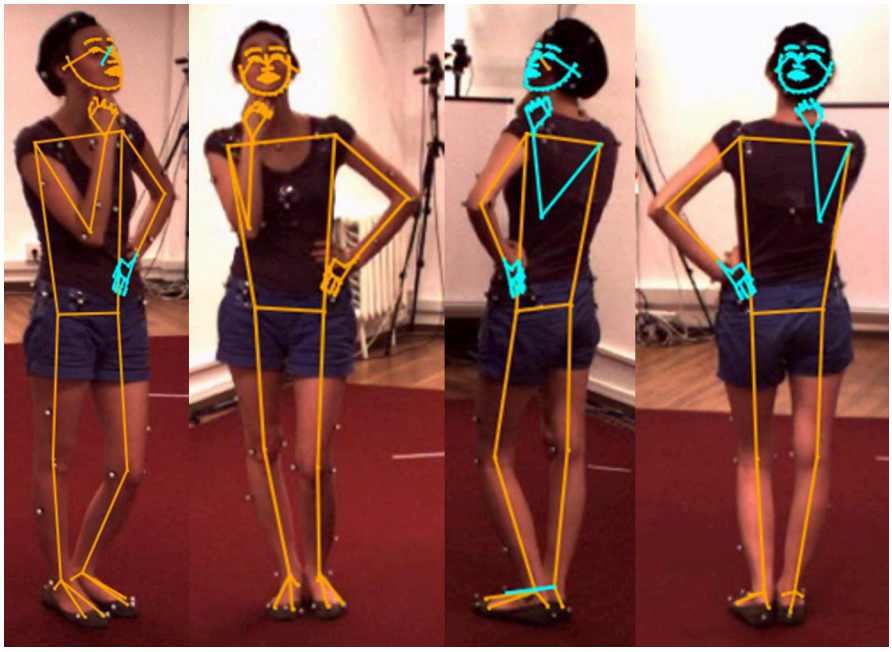

Figure 5.5: Example outputs of the completion network. The orange color denotes the keypoints that were detected by OpenPifPaf. The cyan color shows the missed keypoins by OpenPifPaf but completed by our completion network. The left hand is detected in only 1 view by OpenPifPaf and thus fully predicted by the completion network.

noise to the groundtruth poses with an increasing variance from 5 to 25 pixels, and annotate them as step $t = 1...5$ (step $t = 0$ is the groundtruth). The network learns to predict the pose at step t given the image and the noisier step $t+1$ with a 2D supervision loss.

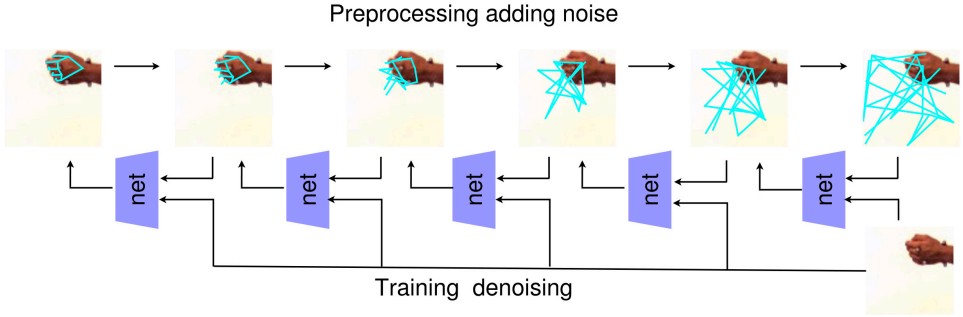

Figure 5.6: Refinement network architecture and training process. Gaussian noise is added to the groundtruth coordinates with increasing variance, and the network is iteratively trained to recover the less noisy coordinates.

We build two small datasets, each consisting of 22,000 non-occluded faces and hands respectively, with their corresponding OpenPifPaf predictions. Each image is resized to 384×384 pixels. We use a random crop of size 224×224 pixels to have the

face and hands located in diverse regions of the images during the training. We split the datasets into training and validation sets with 20,000 images and 2,000 images, respectively.

Quantitatively, the face predictions achieve an average error less than 3 pixels and the hand predictions achieve an average error less than 7 pixels on the validation sets. We show example qualitative results in Figure 5.7.

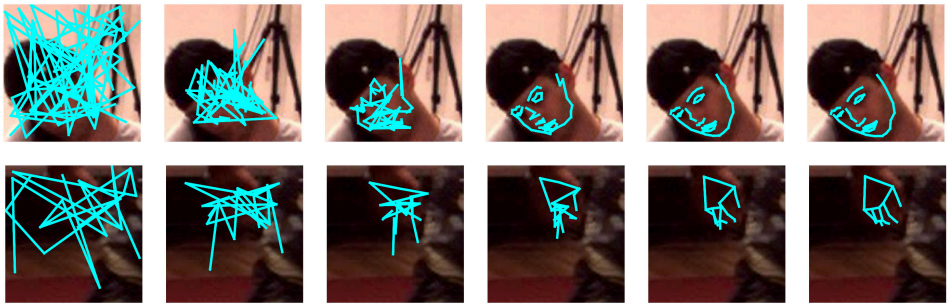

Figure 5.7: Example outputs from the face (top row) and hand (bottom row) refinement networks during inference time. We observe that the predictions almost converge to the correct locations in 5-iteration.

Finally, we run the refinement networks on the 2D-projections of the 3D poses predicted by our completion network. For each 3D skeleton, we project it into the 4 different views. We then crop the regions around the hands and face and denoise the corresponding predictions using the refinement network with 10 iterations to obtain refined 2D poses in each of the 4 views.

Although the refinement network is not always correct due to its training on non-occluded faces or hands, we only need 2 non-opposing views to perform geometric reconstruction. Since bad refinements tend to collapse all keypoints into the same location, we select the two non-opposing views with the highest variance in keypoint positions to avoid disruptions caused by occlusions. Using this method, we obtain 151,036 triplets of 3D whole-body keypoints, corresponding image, and 2D projected keypoints from the original set. Examples of resulting 3D whole-body skeletons and their image-aligned 2D counterparts are shown in Figure 5.8 and Figure 5.9, respectively.

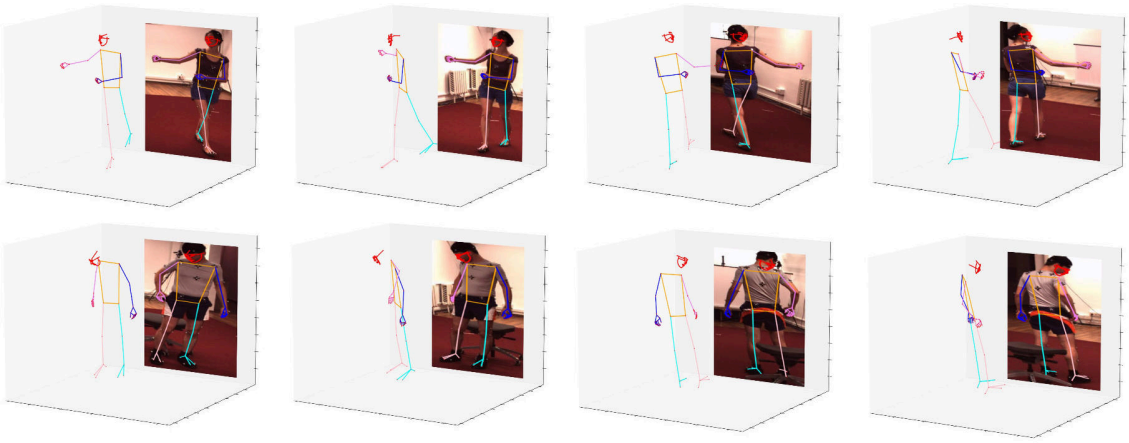

Figure 5.8: Examples of the 3D whole body skeleton. They are visually realistic humans. The strange looking faces (fatter or thinner) in different views are due to viewing artifacts of the default perspective projection.

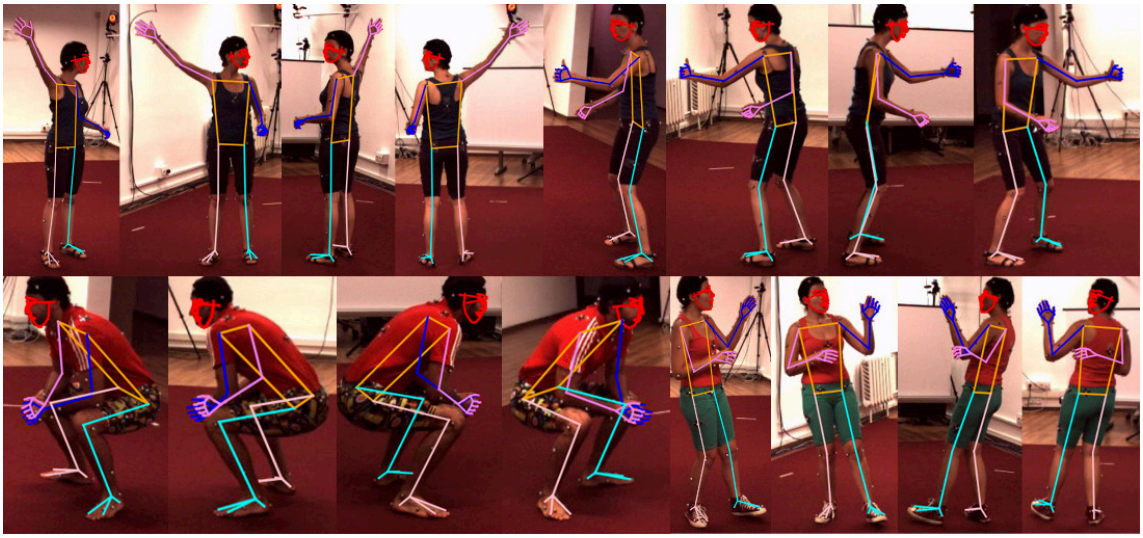

Figure 5.9: Examples of the 3D whole-body skeleton projected in 2D onto their corresponding images. They are visually accurate, though still there are small errors in detail which we do not expect to overcome due to the initial resolution and ambiguity of the images.

5.2.4 Quality assessment

To select the most accurate triplets from our dataset, we reuse the refinement networks and employ a multi-crop strategy that accounts for the variance of the prediction. We project each 3D whole-body skeleton onto all 4 views, and produce four cropped images for each region of interest around the face and hands. The refinement network is run on these 4 crops, and the resulting predictions are aligned with the original prediction to compute the 2D error compared to the original 2D projection. We score the 3D skeletons by averaging the errors of all 4 projected views, and select the 5k lowest error skeletons from each subject of Human3.6M (S1, S5, S6, S7,

S8) to form the 5k $\times$ 4(view) $\times$ 5(subject) = 100k triplets of {image, 2D coordinates, 3D coordinates in camera space} of our 3D whole-body dataset.

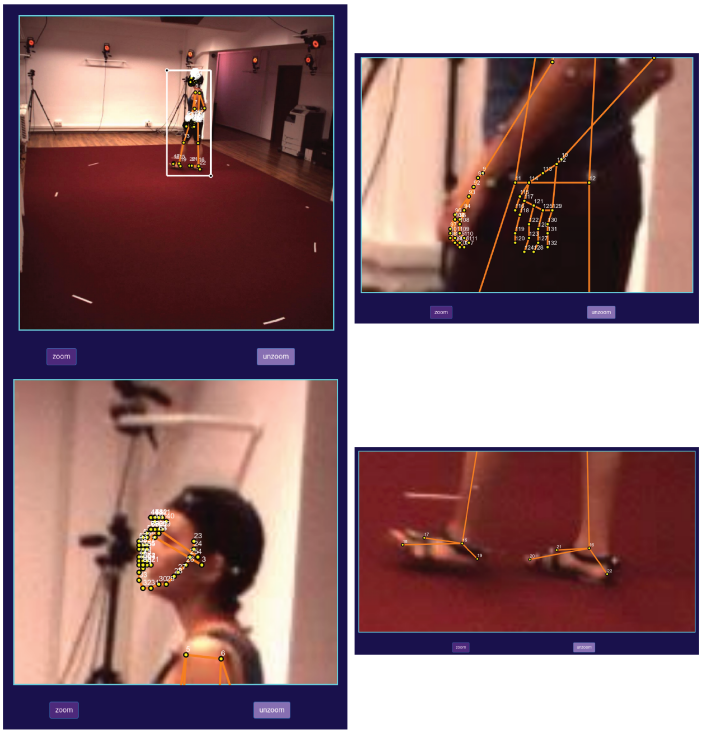

Figure 5.10: Sample screenshots from the annotation interface. Annotators are asked to select area of interest, zoom in on that area and correct the mis-aligned keypoints by drag-drop.

To assess the quality of the H3WB dataset, we conducted a cross-check study on 600 randomly selected images from the dataset. In this study, annotators where presented an image with the 2D projection of the 3D skeleton on top and were asked to manually correct mis-aligned keypoints by drag and drop. The user interface is shown in Figure 5.10, which the user can zoom in to look into details. Using the same multi-view geometry for the first step[13], we reconstruct these corrected skeletons in 3D and compare them to our original skeletons. To validate our process, we show the influence of each step in Table 5.2. The geometric approach produced good results but unfortunately cannot provide a large enough dataset. The completion step allows to obtain all labels but at the cost of degraded accuracy due to lack of alignment as explained in subsection 5.2.2. The diffusion recovers the original accuracy of the geometric approach. 2mm difference is irrelevant given the initial resolution of the images. We obtain a final **average error of 17mm** which is very accurate for such a

13 Initial 3D whole-body dataset with OpenPifPaf

difficult task, and leads to a benchmark which we believe will not be saturated until methods reach around 35mm.

Steps	# keypoints available	3D error (mm)			
		All	Body	Face	Hand
Geometry	48127	14.87	17.72	13.29	15.87
+ Completion	79800	29.31	25.57	26.02	36.67
+ Diffusion	79800	16.98	18.63	15.08	19.16

Table 5.2: Quantitative analysis of each intermediate step in our pipeline.

We also check the distribution of pose per action for H36M and H3WB using the original action labels is shown in Figure 5.11. Apart from *SittingDown*, they are about the same. Quantitatively, we show the standard deviation in mm on average (bold) and for each of the original 17 body joints in Table 5.3 which shows H3WB has slightly lower diversity than H36M, but no collapse.

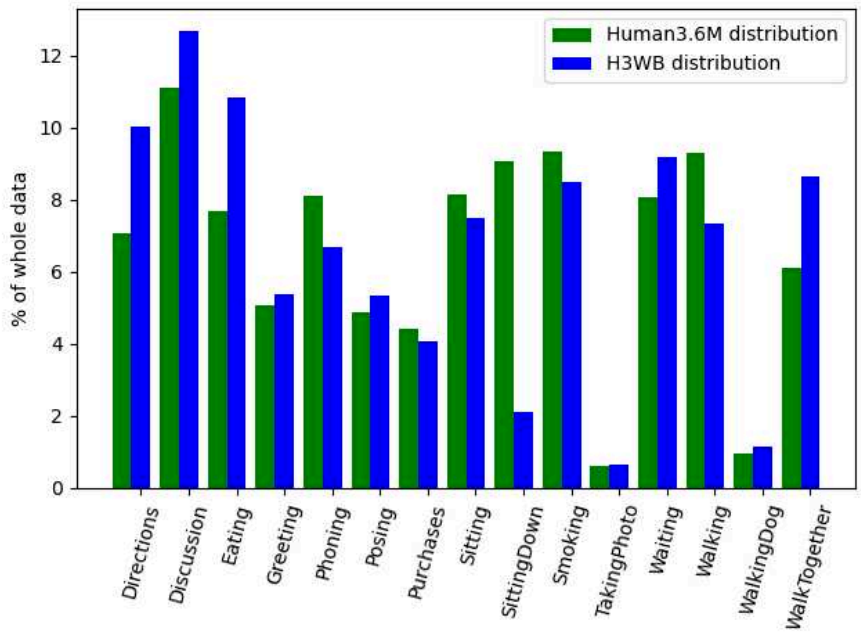

Figure 5.11: Distributions of Human3.6 and H3WB datasets per action class

H36M	**602.7**	540.0	576.3	569.3	578.7	512.8	513.3	527.7	545.3	551.3	552.8	556.5	525.7	518.9	534.8	584.5	624.1	637.4
H3WB	**518.8**	437.9	433.5	444.3	428.9	453.6	422.3	473.1	427.5	505.0	440.2	519.1	419.9	456.1	430.5	462.6	440.3	473.1

Table 5.3: Standard deviation in mm on average (1st column) and for each of the original 17 body joints.

5.2.5 Generalization to other datasets

We run the same dataset making methods with minor differences on videos from CMU Panoptic dataset[14] and MPI-INF-3DHP dataset[15] which provide us another 16k and 26k 3D whole-body skeletons respectively, each of two dataset we use 8 camera views. However, due to the high manual cost for manual-annotated quality assess, we do not do the same quality assess as subsection 5.2.4. Throughout the process, we observe that the things to which we need to pay attention while implementing our dataset making algorithm to different dataset are:

- Different number of camera views, as well as how each dataset store their camera parameters
- The completion network and the diffusion network needs to be fine-tuned to fit the new dataset, even though the pretrained weight from producing H3WB can achieve quite realistic skeletons.

Other than these different points, the remaining code can be the same for different dataset. This shows the strong generalization ability of our method.

[14] [Lab, 2001]

Motion capture database

[15] [Mehta et al., 2017]

Monocular 3d human pose estimation in the wild using improved cnn supervision

5.3 The H3WB benchmark

We use the H3WB dataset to propose a benchmark and the associated leaderboard. We split the dataset into training and test sets. The training set contains all samples from S1, S5, S6 and S7, including 80k {image,2D,3D} triplets. The test set contains all samples from S8, including 20k triplets. The test set labels are retained to prevent involuntary overfitting on the test set. Evaluation is accessible only by submitting results to the maintainers. We do not provide a validation set. We encourage other researchers to report 5-fold cross-validation average and standard deviation.

The corresponding benchmark has 3 different tasks:

1. 3D whole-body lifting from complete 2D whole-body skeletons, or *2D→3D* for short.

2. 3D whole-body lifting from incomplete 2D whole-body skeletons, or *I2D→3D* for short.
3. 3D whole-body skeleton prediction from image, or *RGB→3D* for short.

For each task, we report the following MPJPE (Mean Per Joint Position Error) metrics:

- MPJPE for the whole-body, the body (keypoint 1-23), the face (keypoint 24-91) and the hands (keypoint 92-133) when whole-body is centered on the root joint, i.e. aligned with the pelvis, which in our case is the middle of two hip joints,
- MPJPE for the face when it is centered on the nose, i.e. aligned with keypoint 1,
- MPJPE for the hands when hands are centered on the wrist, i.e left hand aligned with keypoint 92 and right hand aligned with keypoint 113.

To create baselines on each task, we adapt popular methods from the literature by changing the number of keypoints to that of our whole-body dataset. Notice that we keep the training recipes of the original works to avoid over-fitting to this new benchmark. In practice, we perform model selection and hyper-parameters tuning using 5-fold cross-validation.

5.3.1 3D whole-body lifting from complete 2D whole-body keypoints (*2D→3D*)

This task is similar to the standard 3D human pose estimation from 2D keypoints but using whole-body keypoints. The training set contains 80k 2D-3D pairs. The test set contains only a half of all the test samples, i.e. 10k 2D poses[16].

We evaluate 6 methods on this task. SimpleBaseline[17] is a well-established model, consisting of a 6-layer MLP. We propose a modification, replacing the network architecture with an 8-layer MLP, which we call *Large SimpleBaseline* inspired by CanonPose[18]. CanonPose is trained only with 2D supervision. We also adapt CanonPose to work with additional 3D supervision by manually creating 3 fixed

[16] The other half is reserved for the task I2D→3D to prevent access to the missing keypoints.

[17] [Martinez et al., 2017c] A simple yet effective baseline for 3d human pose estimation

[18] [Wandt et al., 2021] Canonpose: Self-supervised monocular 3d human pose estimation in the wild

camera views and rotating the 3D skeletons into the corresponding view before projecting them into 2D, training it with multi-view weak-supervision. Jointformer[19] is a recent transformer-based method. Finally, we report results for the parametric model SMPLify-X[20] by running optimizations on each input sample.

We train SimpleBaseline models using their official training setting. The inputs and targets are normalized by subtracting the mean and dividing by the standard deviation. Similarly, we train CanonPose models following their official training setup where the inputs and targets are centered on the pelvis and scaled by the Forbenius norm. We train the Jointformer model in the two stages as described in their work.

SimpleBaseline and CanonPose models output normalized whole-body keypoints which requires re-scaling at inference. We use statistics from the training set to adjust the test predictions. We calculate a scaling factor using the ratio of 3D to 2D bounding boxes. The formula is: $X_{\text{final}} = X_{\text{unit}} \times \overline{\sigma_{3d}} \times \frac{\sigma_{2d}}{\overline{\sigma_{2d}}}$, where X_{unit} is the normalized prediction, $\overline{\sigma_{3d}}$ is the average size of the 3D training boxes, σ_{2d} is the size of the current 2D box, and $\overline{\sigma_{2d}}$ is the average size of the 2D training boxes.

Since SMPLify-X has 144 keypoints with a different layout, we use interpolation to transform between the WholeBody skeleton and SMPL-X and run SMPL-X's optimization for 2,000 iterations (4 minutes/sample).

Method	All	Body	Face / aligned†	Hand / aligned‡
SMPL-X [Pavlakos et al., 2019]	188.9	166.0	208.3 / 23.7	170.2 / 44.4
CanonPose [Wandt et al., 2021]*	186.7	193.7	188.4 / 24.6	180.2 / 48.9
SimpleBaseline [Martinez et al., 2017c]*	125.4	125.7	115.9 / 24.6	140.7 / 42.5
CanonPose [Wandt et al., 2021] *w* 3D sv.*	117.7	117.5	112.0 / 17.9	126.9 / 38.3
Large SimpleBaseline [Martinez et al., 2017c]*	112.3	112.6	110.6 / **14.6**	**114.8** / **31.7**
Jointformer [Lutz et al., 2022]	**88.3**	**84.9**	**66.5** / 17.8	125.3 / 43.7

Table 5.4: Comparing different methods for 2D→3D on H3WB. Results are shown for the MPJPE metric in mm. Methods with * output normalized predictions. Results of normalized methods are re-scaled using our scaling formula. All results are pelvis aligned, except † and ‡ show nose and wrist aligned results for face and hands, respectively. Sv. is supervision.

We present the results in Table 5.4. SMPLify-X performs the worst, showing that parametric models struggle more than discriminative approaches. SimpleBaseline[21] is a solid method, and Large SimpleBaseline improves its performance further. CanonPose[22] can be improved with additional 3D supervision, but still performs

[19] [Lutz et al., 2022]

Jointformer: Single-frame lifting transformer with error prediction and refinement for 3d human pose estimation

[20] [Pavlakos et al., 2019]

Expressive body capture: 3D hands, face, and body from a single image

[21] [Martinez et al., 2017c]

A simple yet effective baseline for 3d human pose estimation

[22] [Wandt et al., 2021]

Canonpose: Self-supervised monocular 3d human pose estimation in the wild

worse than Large SimpleBaseline. CanonPose also predicts the camera view, and the uncertainty in this prediction can lead to more error. Jointformer[23] achieves the best results among all methods, but still has room for improvement. All methods perform worse on our benchmark than on Human3.6M due to pelvis centering, which creates higher numerical error on extremities like hands and face, the parts that contain most of the whole-body keypoints.

[23] [Lutz et al., 2022]

Jointformer: Single-frame lifting transformer with error prediction and refinement for 3d human pose estimation

5.3.2 3D whole-body lifting from incomplete 2D whole-body keypoints (*I2D→3D*)

We propose a second task where we want to obtain 3D complete whole-body poses from 2D incomplete pose. This task aims to simulate the more realistic case with occlusions and the 2D whole-body detector outputs an incomplete skeleton. We do not provide masks for the training skeletons to allow for online data-augmentation. Instead, we propose a masking strategy as follows:

- With 40% probability, each keypoint has a 25% chance of being masked,
- with 20% probability, the face is entirely masked,
- with 20% probability, the left hand is entirely masked,
- with 20% probability, the right hand is entirely masked.

The second half of the test set (10k 2D) is devoted to this task. The masking strategy is applied only once on the 2D poses of the test set, which are directly provided as incomplete 2D skeletons for fair comparison between methods.

Method	All	Body	Face / aligned†	Hand / aligned‡
CanonPose [Wandt et al., 2021]*	285.0	264.4	319.7 / 31.9	240.0 / 56.2
SimpleBaseline [Martinez et al., 2017c]*	268.8	252.0	227.9 / 34.0	344.3 / 83.4
CanonPose [Wandt et al., 2021] + 3D sv.*	163.6	155.9	161.3 / 22.2	171.4 / 47.4
Large SimpleBaseline [Martinez et al., 2017c]*	131.4	131.6	120.6 / **19.8**	**148.8** / **44.8**
Jointformer [Lutz et al., 2022]	**109.2**	**103.0**	**82.4** / **19.8**	155.9 / 53.5

Table 5.5: Comparing different methods for I2D→3D on H3WB. Results are shown for the MPJPE metric in mm. Methods with * output normalized predictions. Results of normalized methods are re-scaled using our scaling formula. All results are pelvis aligned, except † and ‡ show nose and wrist aligned results for face and hands, respectively. Sv. is supervision.

The results for the I2D→3D task are shown in Table 5.5. All methods perform worse than in the 2D→3D task. SimpleBaseline[24] has low capacity and uses batch normalization that struggles with missing data, resulting in poor performance. The Large SimpleBaseline model, without batch normalization layers, achieves good results for the task's complexity. CanonPose[25] performs poorly due to errors in camera rotation prediction, which are magnified since most of the 133 keypoints are on the face and hands. The addition of 3D supervision partly solves this problem. The transformer-based Jointformer[26] method outperforms others.

5.3.3 3D whole-body pose estimation from a single image (*RGB→3D*)

This task is the standard monocular 3D human pose estimation task extended to whole-body pose estimation. We provide a script to split the original Human3.6M videos into images with our indexing in order to establish image-3D correspondences. The training set contains 80k {image paths,3D} pairs, as well as the 2D bounding box of the human in the image. The test set contains all the test samples, including 20k image paths and their 2D bounding boxes. 2D coordinates are not given in order to avoid collisions with 2D→3D and I2D→3D.

For this task, we run 2 two-stage models and 1 single-stage model. Our first two-stage model uses a Stacked Hourglass Network (SHN)[27] to predict 2D whole-body keypoints and then SimpleBaseline[28] takes 2D keypoint predictions as input and lifts them to 3D coordinates. Similarly, the second two-stage model utilizes Cascaded Pyramid Network (CPN)[29] to output 2D keypoints and then Jointformer[30] lifts the 2D predictions to obtain 3D whole-body poses. For our single-stage model, we modify the last layer of Resnet50[31] to directly output the 3D whole-body keypoints. We regress the 3D whole-body keypoint coordinates using L1 loss.

Results in Table 5.6 show the two-stage *CPN + Jointformer* model obtains the best results. Our simple single-stage method performs better than the two-stage *SHN + SimpleBaseline* model. Learning 2D whole-body keypoints is challenging for SHN as very close keypoints on face and hands may introduce noise to the predicted keypoint heatmaps. The error in the 2D keypoints then makes the lifting task much more challenging. Surprisingly, RGB→3D seems to be harder that the I2D→3D

[24] [Martinez et al., 2017c]
A simple yet effective baseline for 3d human pose estimation

[25] [Wandt et al., 2021]
Canonpose: Self-supervised monocular 3d human pose estimation in the wild

[26] [Lutz et al., 2022]
Jointformer: Single-frame lifting transformer with error prediction and refinement for 3d human pose estimation

[27] [Newell et al., 2016]
Stacked hour-glass networks for human pose estimation

[28] [Martinez et al., 2017c]
A simple yet effective baseline for 3d human pose estimation

[29] [Chen et al., 2017]
Cascaded pyramid network for multi-person pose estimation

[30] [Lutz et al., 2022]
Jointformer: Single-frame lifting transformer with error prediction and refinement for 3d human pose estimation

[31] [He et al., 2015]
Deep residual learning for image recognition

Method	All	Body	Face / aligned †	Hand / aligned‡
RGB→2D+2D→3D:				
SHN [Newell et al., 2016]+SimpleBaseline*	182.5	189.6	138.7 / 32.5	249.4 / 64.3
CPN [Chen et al., 2017]+Jointformer [Lutz et al., 2022]	**132.6**	**142.8**	**91.9 / 20.7**	**192.7 / 56.9**
RGB→3D:				
Resnet50 [He et al., 2015]	166.7	151.6	123.6 / 26.3	244.9 / 63.1
DOPE [Weinzaepfel et al., 2020]	191.3	199.7	187.3 / 66.0	193.3 / 78.2

Table 5.6: Comparing different methods for RGB→3D on H3WB. Results are shown for the MPJPE metric in mm. Methods with * output normalized predictions. Results of normalized methods are re-scaled using our scaling formula. All results are pelvis aligned, except † and ‡ show nose and wrist aligned results for face and hands, respectively.

task. Although there are also missing body parts due to self occlusion, RGB→3D contains more contextual information that should allow to better disambiguate the pose. Compared to 2D→3D and I2D→3D, direct prediction of 3D whole-body pose from images remains thus as a challenging task which we hope this benchmark can help improve over time.

In order to show the importance of training body parts jointly, we evaluate DOPE[32] on our benchmark. Unfortunately, it fails to address occluded body parts only predicts the whole-body keypoints for 35% of the test set. For each missing keypoint, we use the (topological) nearest predicted joint as a proxy. Even so, a disjointed model like DOPE fails to achieve significant accuracy.

[32] [Weinzaepfel et al., 2020]

DOPE: distillation of part experts for wholebody 3d pose estimation in the wild

5.3.4 Qualitative result

Here we show some qualitative in Figure 5.12 outputs obtained by Large Simple-Baseline [33] and Jointformer[34]. Despite slight mis-alignments, the predicted skeletons are realistic.

We also show some examples in Figure 5.13 of a model trained on our H3WB benchmark for the task I2D→3D and applied on COCO dataset[35] with their incomplete 2D wholebody skeleton annotations as input. We can see that even when there are missing points in the 2D input, the model still can predict the 3D wholebody pose accurately. This validates the usefulness of the I2D→3D in real world scenario.

[33] [Martinez et al., 2017c]

A simple yet effective baseline for 3d human pose estimation

[34] [Lutz et al., 2022]

Jointformer: Single-frame lifting transformer with error prediction and refinement for 3d human pose estimation

[35] [Jin et al., 2020]

Whole-body human pose estimation in the wild

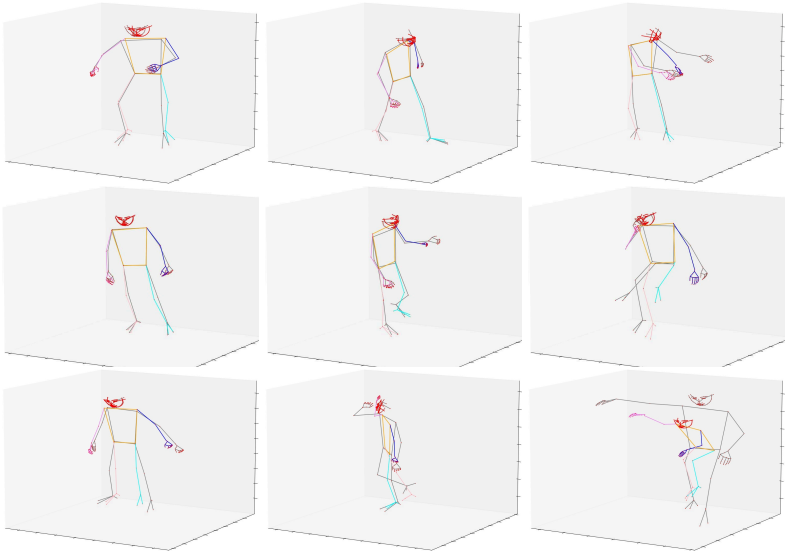

Figure 5.12: Example predictions from Large SimpleBaseline model for 2D→3D (1st row) and I2D→3D (2nd row) tasks. 3rd row shows predictions from Jointformer for RGB→3D task. Colored skeletons correspond to predictions and gray skeletons correspond to groundtruths. First two columns show almost-aligned successful front/side predictions, and the last column shows slightly mis-aligned predictions.

5.4 Limitations

Even though we have achieved promising results, there are still a few limitations of our method that we would like to address in the future:

- The three datasets on which to build our 3D wholebody annotation, Human3.6m[36], CMU Panoptic dataset[37] and MPI-INF-3DHP dataset[38] are mainly indoor dataset. To increase the generalization in the wild setups, we need to apply our dataset creation algorithm to outdoor scenarios.

- To use our dataset creation algorithm, camera calibration parameters must be correctly provided. Our annotation algorithm does not work with a handhold

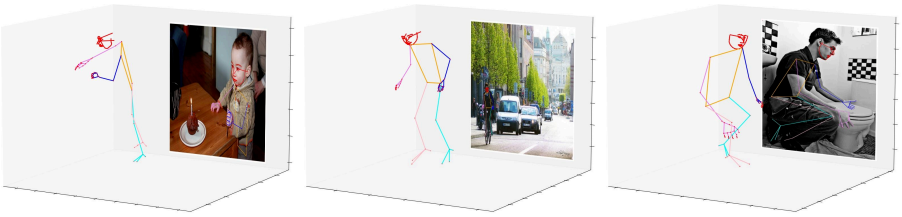

Figure 5.13: Visual examples of lifting on COCO. The labels on the images are the incomplete inputs.

[36] [Ionescu et al., 2014a]

Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments

[37] [Lab, 2001]

Motion capture database

[38] [Mehta et al., 2017]

Monocular 3d human pose estimation in the wild using improved cnn supervision

device and taking several pictures for different views.

- The results we get from running benchmarks from normal 3D human pose estimation are not good enough, which means new models needs to be developed for 3D wholebody tasks to perform better.

5.5 Conclusion

We introduce the H3WB dataset, which extends the Human3.6M dataset with 2D and 3D keypoint annotations for body, face, and hands, containing 100k images with 133 keypoints with an average accuracy error of 17mm. We propose three tasks based on this dataset: 3D whole-body lifting from complete 2D keypoints, 3D whole-body lifting from incomplete 2D keypoints, and 3D whole-body prediction from monocular images. We evaluate several baselines on these tasks and demonstrate promising accuracy, but with room for improvement. Lifting from incomplete 2D skeletons and direct estimation from monocular images remain challenging, and we hope that our dataset and benchmark will spur future research in these areas.

Chapter 6

Interlude: Real-time prototype

In this short interlude, we introduce how we use our previous works to make a prototype, including all steps we claimed in section 3.2, and run it in real time on a computer without a GPU.

6.1 Algorithm

6.1.1 From image to 2D wholebody

The algorithm is run on a computer, we choose to use the computer's own camera as the image captor. In order to increase the performance and make the algorithm run fast, we need to reduce the image to a fairly small size of 192×192 pixels before feeding it into our deep learning network.

Even though in chapter 5 we have trained several backbone networks for tasks from image to 3D, these training datasets are all indoor constrained scenarios and can be problematic if we use them directly in unpredictable real-time scenarios. Thus, we choose to use the pre-trained openpifpaf network[1] to predict 2D wholebody skeleton from the input image, which where trained on in-the-wild images. We used the pretrained weight 'shufflenetv2k16-wholebody' instead of the default weight 'shufflenetv2k30-wholebody', trading a few pixel[2] performance loss with double the running speed.

[1] [Kreiss et al., 2021]

OpenPifPaf:Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association

[2] less than 5 pixel

6.1.2 From 2D to 3D wholebody

As expected, the openpifpaf network only returns the observed keypoints, which means that the 2D wholebody skeleton we obtained is mostly incomplete. In order to lift it into 3D, we choose to use a model trained on task $I2D \rightarrow 3D$.

However, we considered real-time scenarios that there are many cases where people enter or exit the camera screen resulting in only part of their body being in the screen and only this part inside the camera screen can be fed into the network. So during training, we add another masking strategy which has a 50% chance of being used, which randomly chooses between vertical or horizontal, and with a proportion randomly valued between 20% to 80% of the 2D skeleton will be cropped out, leav-

ing only the rest to feed into the lifting network. The other 50% chance are for the same masking strategy as in subsection 5.3.2. This new masking strategy is proved to be very useful, even allowing predictions with very few parts of the observed 2D body.

6.1.3 From 3D wholebody to 3D Ergonova skeleton

Transforming from predicted 3D wholebody skeleton to 3D Ergonova skeleton is not a difficult task, since many of the keypoints are shared between the two skeletons, including keypoints on the head, face, both shoulders, both hips, both hands and both feet. The remainder that requires an algorithm to calculate is listed below:

- Pelvis: COCO wholebody does not have a pelvis joint, but it is essential for calculating the body's flexion and twisting angles. To obtain its position, we predict it using mathematical triangulation (see Figure 6.1).
- For elbows and knees, COCO-wholebody contains its central location, while to calculate supination and pronation we need other orientation information. This orientation is easy to see and calculate if shoulder-elbow-wrist or hip-knee-ankle are not collinear. If it is collinear, we use the orientation of the corresponding wrist or ankle to define the orientation of the elbow or knee.

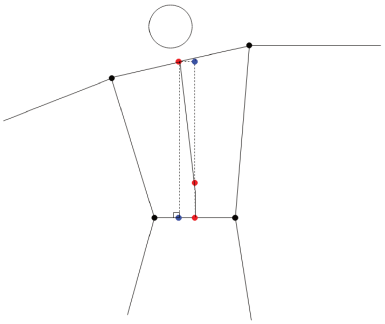

Figure 6.1: The location of the pelvis is calculated as the point where its distances to both shoulders are equal and to both hips are equals, and these two distances follow a pre-defined fixed proportion.

6.1.4 3D Ergonova skeleton to angles

Once the keypoints of the Ergonva skeleton are obtained, we are able to calculate all the angles like flexions, twists, supinations and pronations, etc., that we need from the Ergonova 3D skeleton using geometry algorithms in 3D space.

We show here again the Ergonova skeleton with the keypoint names $(A-Q)$ in Figure 6.2, same as Figure 3.4. For any letter X ($X \in \{A..Q\}$) in this image, if there exists X, X' and X'', then X is on the 'outer-side' of the body

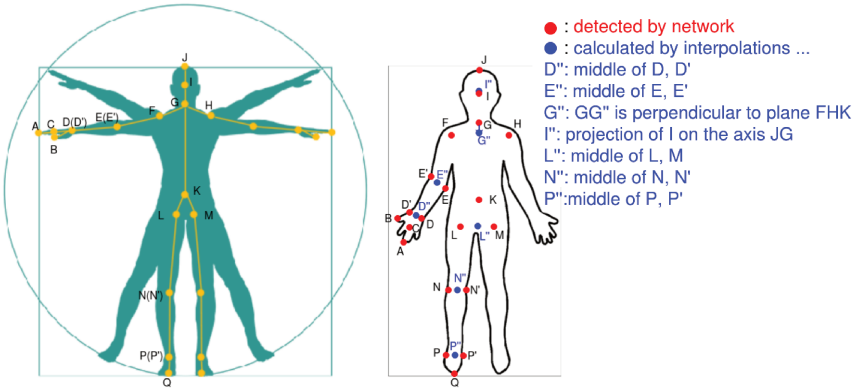

Figure 6.2: The Ergonova skeleton and its keypoint names ($A-Q$) we used to compute angles.

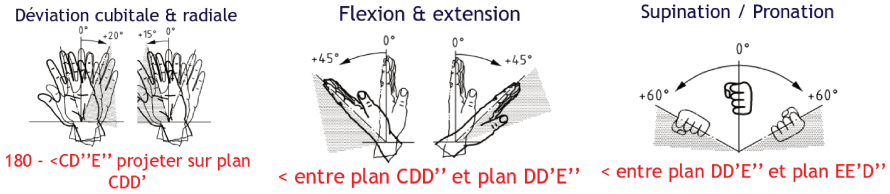

Figure 6.3: The angles around the hand is defined by: (1) the ulnar and radial deviation angle is $\pi-$ the projection of $\angle CD'E''$ on the plane CDD'; (2) the flexion and extension angle is the angle between the plane CDD'' and the plane $DD'E''$; (3) the supination and pronation angle is the angle between the plane $DD'E''$ and the plane $EE'D''$

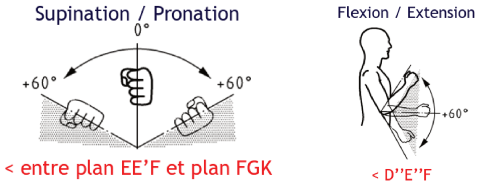

Figure 6.4: The angles around the elbow is defined by: (1) the supination and pronation angle is the angle between the plane $EE'F$ and the plane FGK; (2) the flexion and extension angle is the angle $D''E''F$

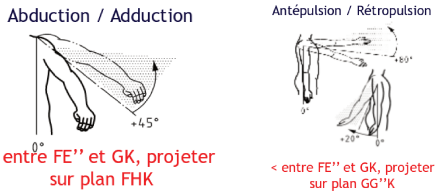

Figure 6.5: The angles around the shoulder is defined by: (1) the abduction and adduction angle is the angle between the vector $\overrightarrow{FE''}$ and the vector $\overrightarrow{GK}$, projecting onto the plane FHK; (2) the antepulsion and retropulsion angle is the angle between the vector $\overrightarrow{FE''}$ and the vector $\overrightarrow{GK}$, projecting onto the plane $GG''K$

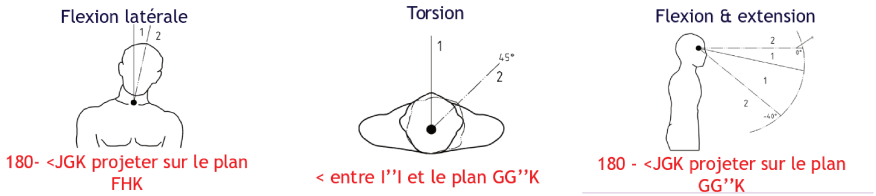

Figure 6.6: The angles around the head is defined by: (1) the lateral flexion angle is $\pi-$ the projection of $\angle JGK$ on the plane FHK; (2) the twist angle is the angle between the vector $\overrightarrow{I''I}$ and the plane $GG''K$; (3) the flexion and extension angle is $\pi-$ the projection of $\angle JGK$ on the plane $GG''K$

Figure 6.7: The angles around the body, or more precisely, the vertebrae, is defined by: (1) the lateral flexion angle is $\pi-$ the projection of $\angle GKL''$ on the plane FHK; (2) the twist angle is $\frac{\pi}{2}-$ the angle between the plane $GG''K$ and the plane KLM; (3) the flexion and extension angle is $\pi-$ the projection of $\angle GKL''$ on the plane $GG''K$

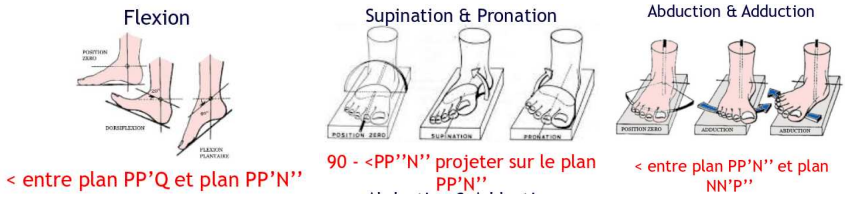

Figure 6.8: The angles around the foot is defined by: (1) the flexion angle is the angle between the plane $PP'Q$ and the plane $PP'N''$; (2) the supintion and pronation angle is $\frac{\pi}{2}-$ the projection of the angle $PP''N''$ onto the plane $PP'N''$; (3) the abductiona and adduction angle is the angle between the plane $PP'N''$ and the plane $NN'P''$

skin, X' is on the 'inner-side' of the body skin, and X'' is the center of rotation of the keypoint. To simplify the calculation, we suppose X'' is the middle point between X and X'. The computed angles are listed on Figure 6.3, Figure 6.4, Figure 6.5, Figure 6.6, Figure 6.7 and Figure 6.8 [3].

[3] Images source: Template squelette 3D_v2_13012021.pptx from Ergonova Conseil.

After obtaining the angles, we compared them to a list of safety zones defined according to data provided by professionals from Ergonova to see if these angles are still safe for the worker in this pose for a long time, and we put those angles that are not in a warning list.

The currently used safe zones are shown in Table 6.1

Part	Angle	Min(rad)	Max(rad)	Min(degree)	Max(degree)
Elbow	supination/pronation	$-\frac{5\pi}{6}$	$-\frac{\pi}{6}$	-150	-30
Elbow	flexion/extension	0	$\frac{2\pi}{3}$	0	120
Shoulder	abduction/adduction	-	$\frac{\pi}{4}$	-	45
Shoulder	antepulsion/retropulsion	$-\frac{\pi}{9}$	$\frac{4\pi}{9}$	-20	80
Head	torsion	$-\frac{\pi}{4}$	$\frac{\pi}{4}$	-45	45
Vertebrae	flexion/extension	$-\frac{\pi}{9}$	$\frac{\pi}{3}$	-20	60

Table 6.1: The table of roughly safe zones we defined according to the data provided by ergonomists from Ergonova.

6.1.5 Render output onto screen

Finally, in order to see the result, we divide the output screen into two, each of which is 480×480 pixels. On the left is the input image as the background and render the 2D prediction on it to show that the 2D prediction is aligned with the image and correct. On the right is the wholebody skeleton lifted in 3D but projected in 4 different views, showing that our lifting is actually in 3D. However, plotting angle numbers on the screen is very ugly, so we choose to use a different color on the skeleton, red color indicating angles in the warning zone and green color indicating angles in safe zone .

6.2 Performance

The whole algorithm runs on a computer with only a CPU at around 1.3 fps. We choose to keep a small skeletal movement that maintains the momentum of speed between previous poses to make the whole performance look smoother. An example screen is shown in Figure 6.9. The algorithm can actually predict and render multi-

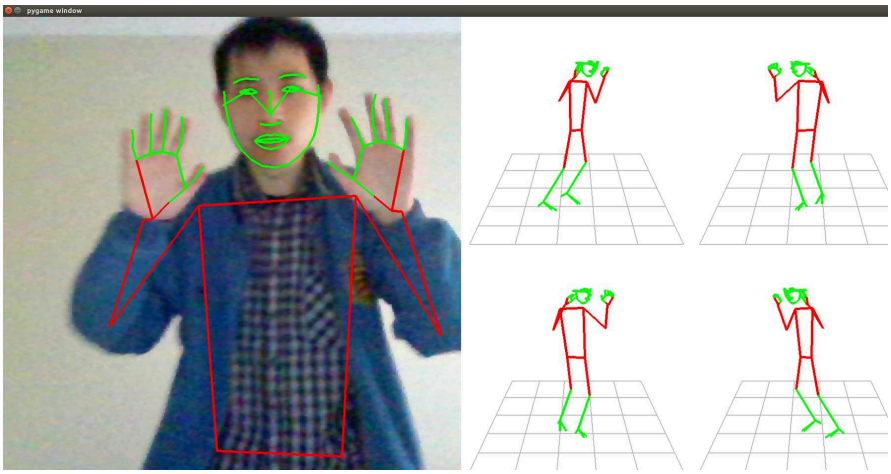

Figure 6.9: An example screen capturing the author of this thesis and rendering the 2D and 3D skeleton on the interface. We can see that even the lower part of the body is not in a 2D image as well as the 2D skeleton, the algorithm still manages to predict the complete 3D skeleton on the right. The bodies are rendered in red because the author bends the body forward, and maintaining this pose is not good for the spine, a problem exists for many people who always sit and work in front of the computer.

person case when there are multiple people on the camera screen. However, the right side will be full of 3D poses which might be a bit chaotic and difficult to study and are therefore not shown here.

Chapter 7

Continuous human motion prediction

In this chapter, we present our project for pose interpolation at extremely low and uneven frame rates, including a methodology for learning and expressing a sequence of human motion in a time-dependent implicit neural representation for use on a Human motion interpolation task where the given input frames are extremely few compared to the whole long sequence.

7.1 Motion interpolation with implicit representation

7.1.1 Implicit representation structure

To elaborate on our problem, given a sequence of 3D human motion with length N: $X_1, X_2, ..., X_N$, we have the ability to observe only a few frames M ($M << N$) denoted as $X_{p_1}, X_{p_2}, ..., X_{p_M}$, where $p_1 < p_2 < ... < p_M$. This setup aims to target the scenario which the motion sequence is incomplete due to corrupted data or human being completely occluded by obstacles like walls and pillars. Our objective is to interpolate the continuous trajectory of the motion sequence using a learned function $\mathscr{G}$ with time t as the input parameter.

To address the problem, we formulate two networks, $\mathscr{F}$ and $\mathscr{G}$, as illustrated in Figure 7.1. Network $\mathscr{F}$ takes the observed 3D human poses (M in total) as inputs and produces the encoded sequence feature z. On the other hand, network $\mathscr{G}$ takes the timestamp t and feature z as inputs, generating the 3D coordinates of the predicted 3D human pose $\hat{X}t$ at time t. Mathematically, this can be expressed as:

$$\mathscr{G}(t, \mathscr{F}(Xp_1, ..., X_{p_M})) = \hat{X}_t \tag{7.1}$$

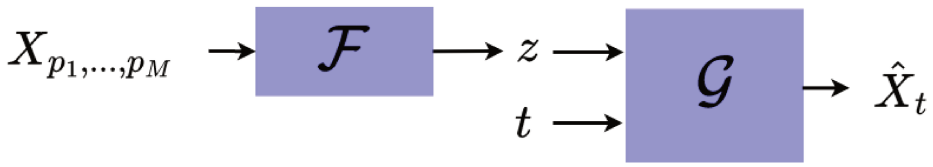

Figure 7.1: The overview of our model. An encoder network $\mathscr{F}$ takes the observed M frame as input, and encode them into latent feature vector z. Network $\mathscr{G}$ takes vector z conditioned on time t as input and outputs the corresponding pose $\hat{X}_t$ at time t.

Here, t represents any real number within the interval of the sequence, where $t \in \{1,2...,N\}$ ideally corresponds to the exact coordinate of that time frame in the ground truth X_t. The notation $\hat{X}$ denotes the predicted pose, contrasting with the ground truth X. This formula is employed during both training and evaluation of our methods.

The main motivation of separate the network $\mathscr{F}$ and $\mathscr{G}$ instead of giving both $Xp_1,...,X_{p_M}$ and t as input to a single network is that in our model, once the input frames $Xp_1,...,X_{p_M}$ are encoded into z, G becomes a function of the single variable t, allowing the expression of the whole movement by only varying the time.

7.1.2 Network design

To maintain simplicity in our model, we propose employing a Multi-Layer Perceptron (MLP) for $\mathscr{F}$ as the encoder. In practice, $\mathscr{F}$ receives as input a batch of $M \times (K \times 3) + 1$ coordinates, where M is the number of input frames and K is the number of keypoints in the skeleton. The $+1$ corresponds to the temporal encoding of input timeframe index. These coordinates are flattened into a vector. Following the approach in siMLPe[1], we utilize 48 residual blocks with the following structure:

[1] [Guo et al., 2023] Back to mlp: A simple baseline for human motion prediction

$$Z_{l+1} = Z_l + \mathrm{LN}(W_l Z_l + b_l) \tag{7.2}$$

Here, Z_l represents the output of block l, W, b_l are the parameters of layer l over temporal dimension of dimension M, and $\mathrm{LN}()$ signifies LayerNorm normalization[2] over spatial dimension of dimention $(K \times 3) + 1$

[2] [Ba et al., 2016] Layer normalization

For the network $\mathscr{G}$, we also opt for an MLP. However, to prevent t from being directly incorporated into a linear function and potentially overshadowed by the much larger feature z, we choose to encode t through linear interpolation of the inputs $X_{p_1}, X_{p_2}, ..., X_{p_M}$. This encoding of t, denoted as $Y(t)$, represents the 3D pose at time t using basic linear interpolation to predict the pose. Mathematically, this is expressed as a continuous function:

$$Y(t) = X_{p_i} + \frac{t - p_i}{p_{i+1} - p_i}(X_{p_{i+1}} - X_{p_i}),\ (p_i \leq t < p_{i+1}) \tag{7.3}$$

To simplify the formula, we assume $p_1 = 1$ and $p_M = N$ in this work, ensuring that we only need to perform interpolation and not extrapolation for times outside the input interval.

Rather than directly approximating X_t, the ground truth at time t, our approach involves predicting the residual between X_t and $Y(t)$, the 3D pose derived from linear interpolation. The overall formulation of our method can be re-expressed as:

$$\mathscr{G}(Y(t), \mathscr{F}(X_{p_1}, ..., X_{p_M})) + Y(t) = \hat{X}_t \tag{7.4}$$

7.2 Training with sequence data

During training, $Y(t)$ and $\mathscr{F}(X_{p_1}, ..., X_{p_M})$ are simply concatenated during the forward pass. We supervise all N frames in the sequence, defining the supervision loss as:

$$\mathscr{L}_{sup} = \frac{1}{N} \sum_{t \in 1,...,N} |\hat{X}_t - X_t| \tag{7.5}$$

Here, we compute the ℓ_1 loss between our prediction at time t and the residual between the ground truth of frame t and the linear-interpolated pose $Y(t)$ for each individual frame.

As we are predicting a continuous function, we have observed that the continuous curve itself can exhibit high-frequency oscillations between the sampled frames, rather than being smooth. To mitigate this effect, we incorporate a regularization loss term $\mathscr{L}_{reg}$ based on the velocity, defined as:

$$\mathscr{L}_{reg} = \frac{1}{N-1} \sum_{t} |(\hat{X}_{t+1} - \hat{X}_t) - (X_{t+1} - X_t)| \tag{7.6}$$

Combining the two components, the final loss is formulated as:

$$\mathscr{L} = \mathscr{L}_{sup} + \alpha \mathscr{L}_{reg} \tag{7.7}$$

Here, α is a constant coefficient set to 0.1 during training.

7.3 Experiments

In this section, we dive into the implementation details and provide an overview of the experimental setups and results.

7.3.1 Datasets and metric

Human3.6M dataset[3]: Human3.6M is a widely used benchmark comprising millions of frames of 3D human poses captured from 7 different actors/actresses performing 15 actions with motion capture equipment. We adhere to the protocols of HisRepIt[4], utilizing S1, S6, S7, S8, and S9 as the training set, and S5 as the test set with 256 test samples per action. Training and testing are conducted on 25 fps videos, equivalent to 1 sample per 2 frames from the original source video.

AMASS dataset: The Archive of Motion Capture as Surface Shapes (AMASS) dataset is a collection of human motion data that amalgamates various motion capture datasets such as CMU Mocap and TotalCapture[5]. AMASS adopts SMPL[6] style data. In our case, we adhere to the practices of HisRepIt and siMLPe[7], utilizing SMPL-H but only incorporating 18 keypoints from the body keypoints while excluding hand information. The train-test split follows the same approach as theirs.

3DPW dataset: The 3D Human Pose in The Wild (3DPW)[8] is a dataset featuring precise 3D annotations for scenarios in the wild, presenting a more challenging environment for the networks. In alignment with HisRepIt, we exclusively utilize the test set of 3DPW with 18 points to evaluate the model trained on the AMASS dataset, assessing the generalization ability of our method.

Evaluation metric: We utilize the Mean Per Joint Position Error (MPJPE) on 3D joint coordinates as the evaluation metric, a common measure in 3D human pose benchmarks. MPJPE calculates the average distances of different joints between the prediction and ground truth. In accordance with standard protocols, we report scores for all datasets with root joint alignment of the prediction and ground truth poses, typically the pelvis joint.

Additionally, we evaluate the standard deviation of the error, along with the minimum (best case) and maximum (worst case) errors among all the test sequences. The

[3] [Ionescu et al., 2014b]

Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments

[4] [Mao et al., 2020]

History repeats itself: Human motion prediction via motion attention

[5] [Trumble et al., 2017]

Total capture: 3d human pose estimation fusing video and inertial sensors

[6] [Loper et al., 2015b]

SMPL: A skinned multi-person linear model

[7] [Guo et al., 2023]

Back to mlp: A simple baseline for human motion prediction

[8] [von Marcard et al., 2018a]

Recovering accurate 3d human pose in the wild using imus and a moving camera

minimum (respectively maximum) is computed by considering the sequence that has the minimum (respectively maximum) error as averaged over its predicted poses. It is important to note that this sequence can be different for each method. This analysis provides insights into the consistency of each method across different scenarios and sequences. Examining the best and worst cases helps to understand the difficulty of the most complex sequence and the potential improvement over linear interpolation for such specific scenarios.

7.3.2 Implementation details

In our practice, we primarily focus on few-shot long-term sequences. Specifically, we set $M = 5$ and $N = 100$, indicating that we have only 5% of known frames within a 4-second sequence sampled at 25 fps. To intensify the task difficulty, we also consider a setup where the remaining $M-2$ known frames, $X_{p_2}, ..., X_{p_{M-1}}$, have non-fixed timestamps. The values of $p_2, ..., p_{M-1}$ are randomly chosen during training. Similarly, during testing, they are randomly selected but with a fixed seed to ensure consistently during evaluation. We employ the Adam optimizer with a learning rate of 10^{-5} and train for 100,000 batches with a batch size of 128. Throughout training and testing, all poses are centered on their root joint. This means the network solely processes the relative position of each joint concerning the root joint to facilitate learning.

For the AMASS and 3DPW datasets, we follow established procedures and disregarding the global rotation in the SMPL model.

7.3.3 Results

This section presents the experiments we conducted to show the effectiveness of our proposed method.

Baseline

The baseline method we use for comparison is the linear interpolation introduced in subsection 7.1.2. This baseline predicts the pose as follows:

$$\hat{X}_t = X p_i + \frac{t - p_i}{p_{i+1} - p_i}(X_{p_{i+1}} - X_{p_i}),\ (p_i \leq t < p_{i+1}) \tag{7.8}$$

The baseline method performs well when the motion itself involves small movements or when the interval between two known frames is small, as observed in an alterntive $M = 10, N = 50$ setup with equal space between input frame, achieving an average error of only 7.7mm. However, in cases where $M << N$ and the input frames become unevenly distributed in the sequence, and if the variation of poses is significantly larger, linear interpolation fails to provide accurate results.

Furthermore, since the methods predict residuals over the linear interpolation, this allows for a clearer assessment of the improvements they bring.

State-of-the-art methods

Given the extreme nature of our setup in terms of deviation from the input, we have chosen and adapted methods from the motion prediction literature for our comparison.

The first model we consider is from siMLPe[9], an multilayer perceptron originally designed for future motion prediction. Given its sequence-to-sequence prediction nature, with an output sequence of the same length as the input sequence, it is agnostic to whether the target sequence is in the future or interpolated between the input sequence, allowing it being directly used for our task.

[9] [Guo et al., 2023]
Back to mlp: A simple baseline for human motion prediction

The second model, inspired by 'History Repeats Itself'[10], employs a graph convolution network. However, since it adopts an auto-regressive prediction strategy, we only use the graph convolution network part to predict all the missing poses simultaneously.

[10] [Mao et al., 2020]
History repeats itself: Human motion prediction via motion attention

In addition, we propose a simple transformer encoder as another baseline. The transformer encoder is trained as a masked auto-encoder[11]. The masking enables it to regress the to-be-predicted poses from the observed ones.

[11] [He et al., 2021]
Masked autoencoders are scalable vision learners

For all these methods, we opt to provide the linear-interpolated sequence $Y(t)$ as input, inspired by the approach in a unified framework[12]. For the transformer, this choice allows us to avoid providing hard positional encoding for the known M frames. Additionally, both of the other two methods output a sequence of the same length as the input, requiring a complete length N sequence as input. Empirically, we found that filling missing points with 0 was less appropriate than feeding the model with linear-interpolated values $Y(t)$. As for the output, all three methods perform residual prediction over the linear-interpolated input $Y(t)$ to approximate the ground truth X_t.

[12] [Duan et al., 2022]

A unified framework for real time motion completion

Human3.6M Result The results of the test on the Human3.6M dataset are presented in Table 7.1. Notably, our model exhibits the smallest average error and standard deviation on this dataset among all methods, indicating its overall superior performance. In the best-case scenario, the transformer achieves the best prediction, but in such a simple motion scenario, even linear interpolation attains excellent performance.

In the worst-case scenario, all methods struggle to surpass the performance of linear interpolation, although our method still achieves the best score. In this complex motion scenario, the regularity of linear interpolation proves advantageous, a characteristic that our velocity regularization function prediction also benefits from.

	Human3.6M				AMASS				3DPW			
Methods	mean	std	min	max	mean	std	min	max	mean	std	min	max
Linear interpolation	71.8	31.7	4.7	229.4	78.8	37.4	**0.6**	367.5	58.3	31.4	**5.0**	205.7
siMLPe [Guo et al., 2023]	66.5	33.7	4.1	271.2	**73.3**	**34.2**	1.0	330.5	**52.6**	**24.4**	5.5	**180.7**
Transformer [He et al., 2021]	70.5	33.2	**3.6**	263.0	78.9	37.2	0.7	322.1	58.5	31.2	5.5	198.4
HisRep [Mao et al., 2020]	67.6	31.2	4.8	281.1	77.7	35.0	1.9	318.7	58.3	29.7	6.0	203.0
PIUS (ours)	**64.0**	**30.2**	4.5	**222.1**	73.5	34.4	1.2	**301.7**	55.4	27.3	6.1	184.7

Table 7.1: Quantitative comparison with the state-of-the-art methods on Human3.6M [Ionescu et al., 2014b], AMASS [Mahmood et al., 2019] and 3DPW [von Marcard et al., 2018a] datasets. Models are trained by sampling **5 random frames** out of 100. Results are presented on the MPJPE metric in mm. *min* (respectively *max*) score corresponds to the sequences with the lowest (respectively highest) average error. Best results are **boldfaced**, as well as second best results are underlined.

AMASS and 3DPW Result The results of the test on the AMASS dataset and 3DPW dataset are displayed in Table 7.1. Our method demonstrates competitiveness

among all methods. While all methods are closer to linear interpolation, indicating that the sequences may contain easier motions compared to the Human3.6M datasets, our approach remains robust. Notably, for the worst sequence, we are able to achieve a significant improvement over the baseline.

7.4 Details studies

7.4.1 Other Scenarios

In addition to the main result section, where we predominantly introduce our method in a few-shot random frame setup with $M = 5$ and $N = 100$, we also explored two different setups to assess their impact on performance.

Uniform Sampling In this scenario, instead of randomly sampling the input frames, we opt for equally distributing the interval between input frames to assess its impact on performance. Intuitively, this should be an easier task, as it minimizes the chances of losing complex motion between distant input frames without any a priori knowledge about the moment in time when such motion occurs. In this setup, among the 100 total frames, the 5 input frames are fixed at frames 1, 25, 50, 75, and 100.

	Human3.6M				AMASS				3DPW			
Methods	mean	std	min	max	mean	std	min	max	mean	std	min	max
Linear interpolation	57.0	26.4	3.9	**186.7**	73.6	38.3	**0.6**	266.3	51.8	31.7	4.3	196.2
siMLPe [Guo et al., 2023]	48.6	25.6	3.9	199.3	56.9	33.7	0.8	267.7	**38.9**	**22.2**	**4.2**	**187.4**
Transformer [He et al., 2021]	57.0	26.4	3.9	**186.7**	73.6	38.2	0.7	266.3	51.8	31.7	4.3	196.2
HisRep [Mao et al., 2020]	53.0	24.7	4.1	192.3	57.8	**29.7**	1.1	**264.9**	44.3	22.5	5.4	192.4
PIUS (ours)	**47.9**	**24.4**	**3.7**	195.6	**52.0**	35.4	1.3	265.8	40.4	23.1	4.7	194.5

Table 7.2: Quantitative comparison with the state-of-the-art methods on Human3.6M [Ionescu et al., 2014b], AMASS [Mahmood et al., 2019] and 3DPW [von Marcard et al., 2018a] datasets. Models are trained by sampling **5 fixed frames** out of 100. Results are presented on the MPJPE metric in mm. *min* (respectively *max*) score corresponds to the sequences with the lowest (respectively highest) average error. Best results are **boldfaced**, as well as second best results are underlined.

The corresponding results on Human3.6M are presented in Table 7.2. As observed, the task is indeed easier, with the linear interpolation baseline achieving a significantly better average. Nonetheless, our method is able to surpass that baseline by more than 10mm.

	Human3.6M				AMASS				3DPW			
Methods	mean	std	min	max	mean	std	min	max	mean	std	min	max
Linear interpolation	44.9	22.4	3.5	**162.6**	53.5	29.1	**0.5**	**194.9**	39.8	25.2	2.9	**166.7**
siMLPe [Guo et al., 2023]	38.5	**19.5**	2.3	174.0	**34.0**	**21.3**	0.6	205.5	**30.3**	**19.2**	**2.5**	180.6
Transformer [He et al., 2021]	42.9	21.9	**2.0**	171.3	53.2	29.9	**0.5**	235.5	38.3	25.1	3.1	173.2
HisRep [Mao et al., 2020]	43.3	21.5	2.6	171.4	51.0	27.6	**0.5**	201.9	37.7	23.8	3.3	169.6
PIUS (ours)	**38.1**	20.2	2.7	181.4	50.5	27.4	0.8	220.3	37.5	24.0	**2.5**	184.3

Table 7.3: Quantitative comparison with the state-of-the-art methods on Human3.6M [Ionescu et al., 2014b], AMASS [Mahmood et al., 2019] and 3DPW [von Marcard et al., 2018a] datasets. Models are trained by sampling **10 random frames** out of 100. *min* (respectively *max*) score corresponds to the sequences with the lowest (respectively highest) average error. Results are presented on the MPJPE metric in mm. Best results are **boldfaced**, as well as second best results are underlined.

Surprisingly, it appears to be challenging to improve over the linear interpolation baseline for the most difficult sequence in this setup as it is significantly improved over the random sampling setup, showing that the randomness is indeed having an big impact of the difficulty of some sequences.

The results for AMASS and 3DPW are displayed in Table 7.2. On AMASS, the task does not seem significantly easier than in the random sampling setup. However, while our method struggled to improve over the baselines in the random sampling case, here, it outperforms linear interpolation by a significant margin. These results hold on 3DPW, even though the model was trained on AMASS only (zero-shot setting).

Longer Input Length The second setup involves a random frame arrangement but with more input frames, i.e., with $M = 10$ instead of $M = 5$. Intuitively, this should be an easier task, as linear interpolation becomes more challenging to beat when the chances of capturing complex motion between two input frames are reduced.

The corresponding results for Human3.6M are presented in Table 7.3. Indeed, the linear interpolation baseline shows a significant improvement over the $M = 5$ setup, indicating that shorter timespans are indeed easier to predict. However, our method still manages to outperform all others by more than 6mm. Surprisingly, the worst sequence appears to be better predicted by linear interpolation, suggesting that in this case, learning methods may hallucinate oscillations.

The results for AMASS and 3DPW are displayed in Table 7.3. In this setup, all methods perform similarly, with the notable exception of siMLPe, which is able to

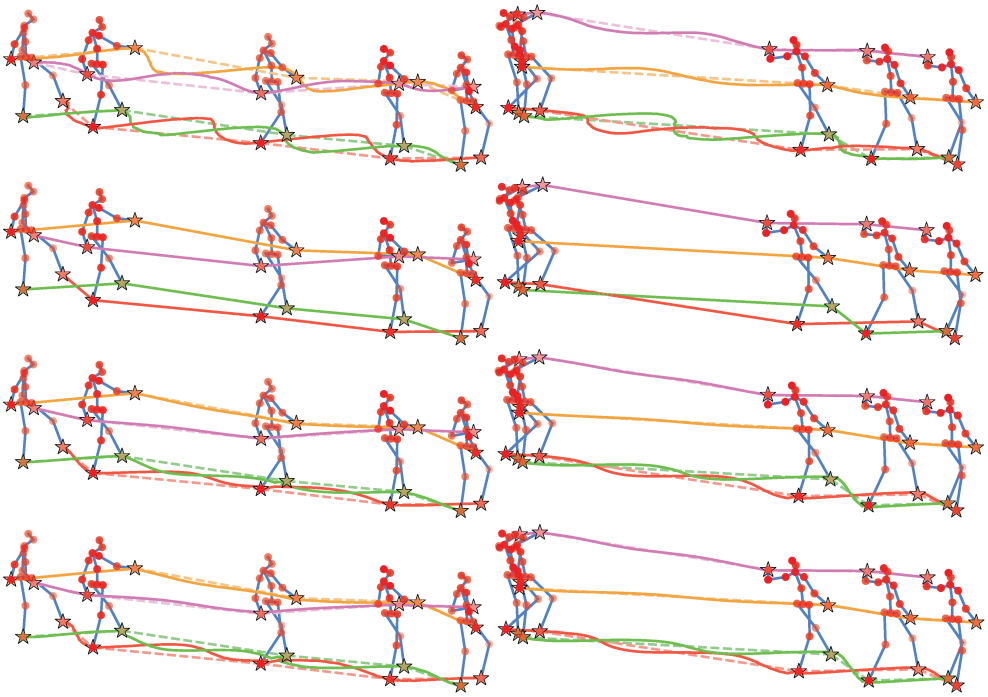

Figure 7.2: **Qualitative results.** Example sequences of 3D predictions in the $M = 5, N = 100$ setup. We present qualitative results for siMLPe [Guo et al., 2023] **(top row)**, Transformer [He et al., 2021] **(second row)**, HisRep [Mao et al., 2020] **(third row)** and our method PIUS **(bottow row)**. The dashed lines correspond to Linear interpolation. 4 different colors indicate the 4 most variant joints within the sequence. Linear interpolation trajectory is not visible due to its high overlap with Transformer's predicted trajectory. Linear interpolation's simplistic approach results in straight-line trajectories that lack realism. On the other hand, siMLPe [Guo et al., 2023] produces oscillated trajectories that deviate from natural motion patterns, introducing unrealistic fluctuations in the predicted trajectories. Together with HisRep [Mao et al., 2020], our method PIUS is able to predict both smooth and realistic motion trajectories.

outperform all others. This might be explained by the structure of siMLPe, which is well-suited for denser sequences.

7.4.2 Qualitative results

Here we present qualitative results of 3D trajectories from the Human3.6M dataset in Figure 7.2. These trajectories correspond to predictions of a 4-second sequence ($N =$ 100 frames) with only $M = 5$ randomly selected input frames. The blue skeletons represent the pose given as input. The dashed lines correspond to linear interpolation, while the solid lines represent the predictions of each method.

As seen in Figure 7.2, left sequences, the transformer-based method closely re-

sembles linear interpolation, while other methods correctly predict the walking pattern. It also appears that the siMLPe method is hallucinating exaggerated motion, which may explain its poor performance on the most challenging sequences.

The sequences on the right side of Figure 7.2 demonstrate the same pattern, with increased hallucinated oscillations on the hands and feet for siMLPe compared to other methods. In this specific case, the Mean Per Joint Position Error (MPJPE) of siMLPe is 164mm, compared to 128mm for our method (and 126mm for HisRep and 163mm for the transformer). This indicates that siMLPe is indeed hallucinating exaggerated oscillations. This is surprising since two frames are given closely at the beginning of the sequence, which could provide an estimation of the angle and speed of the motion.

7.4.3 High frequency prediction

Since we model the motion of the sequence as a continuous function $\mathscr{G}$ that can take any time step t as input, we can resample a given input sequence at any framerate, including very high ones.

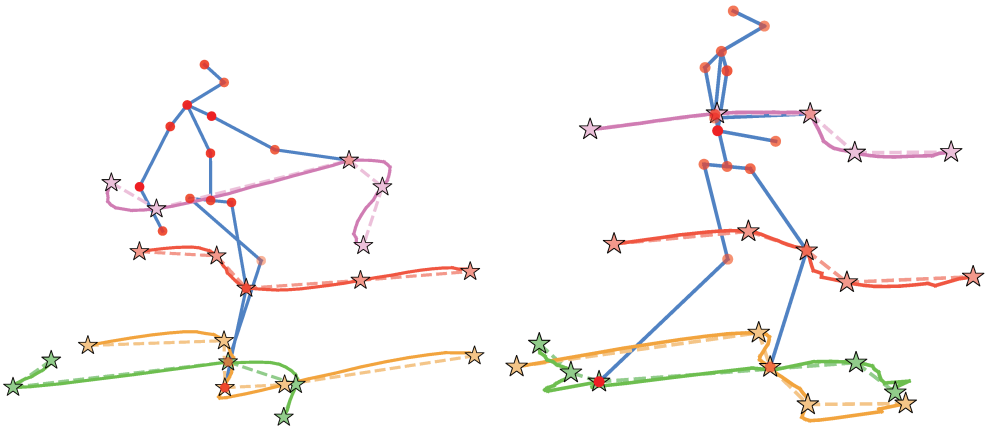

Figure 7.3: **Qualitative results on high FPS prediction.** We present motion trajectories obtained by predicting sequences around an input frame at high FPS. The input frames are depicted as solid blue skeletons. 4 different colors indicate the 4 most varying joints within the sequence. Traditional discrete methods can only generate predictions at their given-frequency (starred timestamps). Thus these methods linearly interpolates for timestamps in between (shown by dashed lines). Our approach is able to predict continuous smooth motion trajectories at high FPS.

In this experiment, we take the model trained in the $M = 5, N = 100$ setup with random frame sampling and use it to resample 2 seconds of a 25fps sequence by 10

times the original frequency (250fps).

The resulting 3D trajectories are shown in Figure 7.3. As observed, the baseline linear interpolation produces jagged motion, whereas our interpolation provides much smoother trajectories. This is particularly noticeable for extremities like the hands or feet.

Note that the choice of the resampling frequency is arbitrary, and we could resample to over 100 fps if needed. However, we found the difference with 25 fps to be difficult to demonstrate on paper.

7.5 Operator valued kernel strategy

Before setting on the current solution, we also experimented with a network $\mathscr{F}$ that can directly predict the parameter of the function $\mathscr{G}$ instead of predicting a feature z passed to $\mathscr{G}$, leaving $\mathscr{G}$ truly a function that only depends on time t. This strategy is called operator valued kernel[13]. We test this strategy on a future motion prediction scenario, which the model structure is shown in Figure 7.4.

[13] [Kadri et al., 2016]

Operator-Valued Kernels for Learning from Functional Response Data

In this future motion prediction scenario, the last known frame p_M is not the last frame of the sequence N, which we need to do extrapolation into future. However, most studies of future motion prediction take into account of all frames in the past, and we use a commonly studied setup for motion prediction, which $N = 75$, $M = 50$ and $p_1, ..., p_M = 1...50$

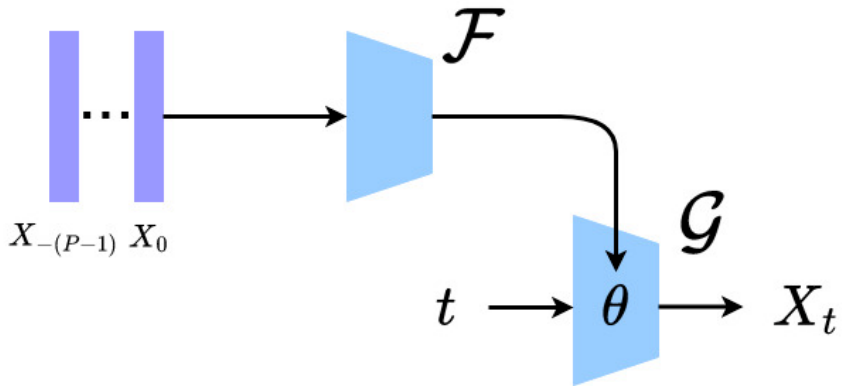

Figure 7.4: The operator valued kernel model structure. An encoder network $\mathscr{F}$ takes the observed past M frames as input, and directly predicts the parameter θ_G of the network $\mathscr{G}$. Network $\mathscr{G}$ takes only the time t as input and outputs the corresponding pose $\hat{X}_t$ at time t.

Under this setup, the network $\mathscr{F}$ has to deal with much more data than motion interpolation task, and we therefore set it to 48 layers instead of 6. On the other hand, in order to prevent the output layer of $\mathscr{F}$ to overload due to predicting too many values, $\mathscr{G}$ has to be small, and we set it to a single 2 layer MLP: the input time t is sinus-encoded into 32 dimensions, and the first layer transform 32 parameters to 128, and the second layer transform 128 parameters to $K \times 3$ with K the number of keypoints. We use the same training loss as motion interpolation task, except we supervise the whole sequence of N frames, including both the past and future frames, expecting the network can both reconstruct the past and predict the future, as well as maintaining continuity over time.

The scores are shown in Table 7.4. This method fails to improve our best models for predicting the future. However, we apply the same evaluation metric but on $t = -49, .., -1, 0$, the past 50 input frames, and we obtained that the average MPJPE loss is 7.4 mm on all these input frames, which means they are overfitted. Visually we plot the graphs of such movements and we can see examples in Figure 7.5.

	MPJPE(mm)↓							
Times(ms)	80	160	320	400	560	720	880	1000
Res-RNN [Martinez et al., 2017b]	25.0	46.2	77.0	88.3	106.3	119.4	130.0	136.6
convSeq2Seq [Li et al., 2018]	16.6	33.3	61.4	72.7	90.7	104.7	116.7	124.2
LTD-50-25 [Mao et al., 2019b]	12.2	25.4	50.7	61.5	79.6	93.6	105.2	112.4
LTD-10-10 [Mao et al., 2019b]	11.2	23.4	47.9	58.9	78.3	93.3	106.0	114.0
Hisrep [Mao et al., 2020]	10.4	22.6	47.1	58.3	77.3	91.8	104.1	112.1
MSR-GCN [Dang et al., 2021]	11.3	24.3	50.8	61.9	80.0	-	-	112.9
ST-DGCN-10-25 [von Marcard et al., 2018c]	10.6	23.1	47.1	57.9	76.3	90.7	102.4	109.7
siMLPe [Guo et al., 2023]	9.6	21.7	46.3	57.3	75.7	90.1	101.8	109.4
Ours	14.5	27.7	52.2	62.8	80.1	92.9	103.3	110.9

Table 7.4: Future motion prediction result on Human3.6M dataset.

To avoid overfitting on the past frames we propose to steer the prediction towards a 'groundtruth' model obtained on the entire sequence.

We first want to check if such groundtruth model exist. For each test sequence, we use an optimization algorithm to overfit parameter of $\mathscr{G}$ on each sequence containing both the past and future frame, and we called it θ^*_{P+F}, the optimal weight of $\mathscr{G}$ on 'P'ast and 'F'uture. We show in Table 7.5 that the optimal θ^*_{P+F} with 100k iteration of optimization can reach single-digit error, thus such θ^*_{P+F} exist.

We then calculate this θ^*_{P+F} model for each training sequence during the training

Figure 7.5: The first row contains 2D samples and the second row contains the corresponding 3D samples. The orange poses are pose X_0 at $t = 0$. The light blue curves are the groundtruth movement curve from $t = -49$ to $t = 0$, while the dark blue curve is our prediction from the past reconstruction. We can see the dark blue curves almost cover the light blue curves of the image, in both 2D and 3D.

	MPJPE(mm)↓							
Times(ms)	80	160	320	400	560	720	880	1000
θ^*_{P+F} (50k iter)	39.7	51.0	70.2	77.1	88.5	99.3	110.7	122.5
θ^*_{P+F} (100k iter)	6.1	7.0	7.5	7.6	7.4	6.7	5.7	13.8

Table 7.5: Test if optimal θ^*_{P+F} exist

as target and see if $\mathscr{F}$ can learn θ^*_{P+F}. With a lot of tunings, unfortunately, the score is not better at all, shown in Table 7.6.

	MPJPE(mm)↓							
Times(ms)	80	160	320	400	560	720	880	1000
θ^*_{P+F} sup.	15.4	28.1	54.4	66.7	84.5	98.3	108.6	115.4

Table 7.6: Best score we get with supervision of θ^*_{P+F}

We then study whether learning to predict optimal weight from coordinates is too difficult, and whether it might be easier to learn from one weight to another weight. We calculate θ^*_P, the optimal weight of $\mathscr{G}$ only optimized on the past, for each train and test sequence, and we modify $\mathscr{F}$ to predict θ^*_{P+F} from θ^*_P.

Since $\mathscr{G}$ has two layers of weight of size 32×128 and $128 \times (K \times 3)$ respectively,

the optimal weight θ^*_P and θ^*_{P+F} are are stored in a matrix of size $128 \times (32 + K \times 3)$. Considering that alternating order between the neurons in the hidden layer of size 128 will not affect the output, we can practice a matching strategy on θ^*_{P+F} on the first dimension to make it easier to learn from θ^*_P. As such, we use the python package lapsolver (Linear Assignment Problem solver) as matching algorithm to match each row of θ^*_{P+F} to the closest θ^*_P. Once the matching is done, we carry out a utility analysis of the matching. We use linear interpolation between θ^*_P and θ^*_{P+F} to obtain 50 intermediate θ, and use them as $\mathscr{G}$ to see how error varies. The result is shown in Figure 7.6 where clearly the matched weight has smaller error in between, leading to a weight θ^*_{P+F} potentially easier to learn.

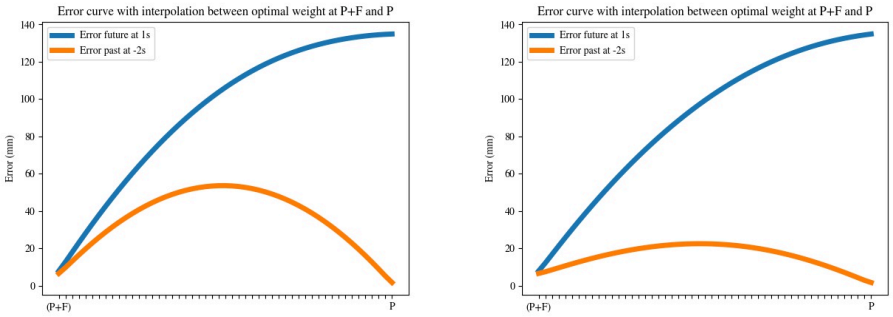

Figure 7.6: The error score (in mm) using the interpolated weight before (left) and after (right) matching. Since alternating order between the neurons in the hidden layer will not affect the output, the scores are the same at both extremities, which mean θ^*_P and θ^*_{P+F}. However, the one with matching shows a lower intermediate error in between, shown an potentially easier path to go from θ^*_P to θ^*_{P+F}.

We also calculate θ^*_P and θ^*_{P+F}, and examine the nearest neighbor of each θ^*_P before and after matching, to see if the corresponding θ^*_{P+F} is always the closest to θ^*_P. We plot the nearest neighbor in a heatmap shown in Figure 7.7, clearly showing each pair of $(\theta^*_P, \theta^*_{P+F})$ has closest distances within the pairs after matching, whereas they are not before matching.

We thus train a network $\mathscr{F}$ to learn to predict matched θ^*_{P+F} from θ^*_P. The best score is shown in Table 7.7

Unfortunately, running optimization algorithm to obtain θ^*_P and θ^*_{P+F}, as well as matching strategy are all very time consuming. Without much progress in terms of score and results, we choose to postpone this direction, hoping that it can be resumed

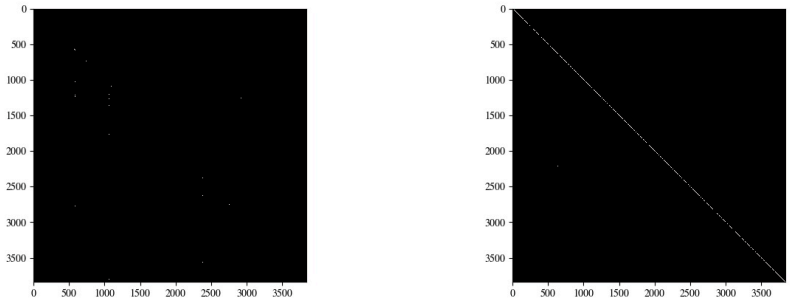

Figure 7.7: The nearest neighbor heatmap before (left) and after (right) matching on approximately 4000 sequences in test dataset. A clear pattern is that it is chaotic before matching, but almost diagonal after matching, meaning the nearest neighbor of θ_P^* among all θ_{P+F}^* in the dataset is always it matched correspondent one.

	MPJPE(mm)↓							
Times(ms)	80	160	320	400	560	720	880	1000
$\theta_P^* \rightarrow \theta_{P+F}^*$	22.5	39.4	61.0	68.4	82.7	97.1	104.7	109.7

Table 7.7: Best score achieved by training $\mathscr{F}$ predicting θ_{P+F}^* from θ_P^*

once we have some breakthrough ideas.

7.6 Limitations

We list here the limitations of our method that we would like to address in the future:

- Our model does not always achieve the best score among all different setups and datasets, which means a better model needs to be adapted in order to further increase the performance, otherwise we cannot say that we are better than other methods based on qualitative results.

- Even though the network $\mathscr{G}$ is designed as a function depending on time t with sinus encoding, which should be relatively smooth over time, the predictions can still be full of twists. This can be partly solved by tuning the sinus encoding numbers or frequencies, but we do not having a perfect solution for this problem yet.

- We did not explore additional constraints or tasks to facilitate motion interpolation, such as action class conditioning, incomplete motion completion, or

noisy motion denoising. These tasks might be helpful to each other.

7.7 Conclusion

We address the challenge of motion interpolation in an extremely low frame rate, where only 5% of the input poses are provided. Additionally, we consider the scenario in which the given input poses are randomly sampled to account for the varying computation time of a pose estimation method producing these inputs. To address this problem, we propose a novel approach that models human motion as a continuous function implemented by a neural network, akin to neural implicit representations.

We perform a comprehensive comparison of this approach with state-of-the-art motion prediction methods on three popular datasets, demonstrating significant improvements over baselines in most cases. The visualizations illustrate that our method generates smooth trajectories compared to simple linear interpolation, without introducing exaggerated motion.

Chapter 8

Conclusion

What we did

3D human pose estimation is indeed a subject widely researched across different models and scenarios, and yet many new things can still be explored. Since this topic is related to humans, it means that it also has a wide potential area of application.

In this thesis, we first design an algorithm which allows to generate synthetic 3D human skeletons on the fly during the training of 2D to 3D human pose lifting task, following a Markov-tree type distribution which evolve over time to create new unseen poses. The K-NN based precision-recall evaluation metric shows the similarity of our generated skeleton with real human poses. We show that the parent-child angles under our designed spherical coordinate system have a clear distribution pattern according to real datasets, which is a potential practical utility for further studies.

We then introduce the H3WB dataset, which extends 2D and 3D keypoint annotations for body, face, and hands with 133 keypoints, as well as three tasks for 3D wholebody pose estimation based on this dataset: from complete 2D keypoints, from incomplete 2D keypoints, and from a single image. We argue that the proposed task for 3D wholebody pose estimation from incomplete 2D wholebody human pose best fits the real world scenario, in which many cases of occlusions may occur. We also find that the wholebody annotation allows wider areas of application, not only for the angle calculation that our thesis requires, but also potentially to help dealing with body action, hand action and facial expression as a whole for real world interactions.

We also address the challenge of motion interpolation at extremely low frame rate, by proposing a new approach that models human motion as a continuous function implemented by a neural network, akin to neural implicit representations. By expressing motion as a function of time, we allows the interpretation of the whole motion at any fps in a smooth manner. We expect this to be practical for analyzing human poses in video, where human motion detected in video can be jagged and inconsistent.

We finally implement a real time prototype of 3D wholebody pose estimation from camera-captured real-time image, running on a CPU-only computer. We use a pretrained 2D wholebody estimation model as well another 2D to 3D lifting model

trained from incomplete 2D wholebody data. The estimated 3D skeleton is then transformed into the Ergonova skeleton model to perform angle computation and dangerous detection.

To go further

In order to dive into 3D human pose estimation further, we can try to improve the performances in our current projects by dealing with limitations of each project. There are also quite a few things we can do by merging the ideas derived from previous works.

In short term, **Synthetic wholebody generation** can be an interesting subject. Since now we have a well constructed 3D wholebody skeleton dataset, we can take design a markov tree to define the relationship tree between the keypoints from the wholebody skeleton, and use the distribution according to the distribution graphs to generate more accurate wholebody 3D poses for synthetic training. However, the design of such hierarchical markov tree should be carefully finetuned since they are not as straight as simple skeletons with around 20 keypoints. Other than this, **Wholebody motion interpolation** can be easily studied just by replacing the simple skeletons to wholebody skeleton. The advantage of using wholebody skeleton is that motion interpolation should be easier with the wholebody skeleton, since there are now more constraints on the keypoints with each other. Unfortunately, linear interpolation methods does not work at all for wholebody skeleton, especially on the hands and face, making the realistic motion interpolation more essential for wholebody skeleton model.

In long term, we can study on **Motion interpolation with other auxiliary tasks**. For motion interpolation, incorporating body constraints as auxilary tasks could be explored, ensuring that the interpolated trajectory preserves certain body invariant (e.g., bone length, symmetry). Apart from this, motion action recognition can be performed simultaneously with motion interpolation, which also provide constraints to the learnt motion sequence. These all need some big modification of current structure. However, this topic has already been studied by many existing works like using diffusion model[1] or using large language model[2], making it a very competitive but hot topic. We also want to **Improve our prototype**. The current prototype we have

[1] [Tevet et al., 2023]

Human Motion Diffusion Model

[2] [Jiang et al., 2023]

MotionGPT: Human Motion as a Foreign Language

made is still a lab version, which is far from being used or published. We may have a few update ideas: Algorithmically, we need a more precisely-defined safe zone for the angles instead of our roughly estimated safe zone. A possible way is to use the distribution graphs of angles synthesized from the parent-child relation maps to decide the safe zone. Other than that, we still want to improve the prediction speed, and also make it run as an application on handphones, a real tool to help daily working people, which we need a lot works to do.

Bibliography

[Akhter and Black, 2015] Akhter, I. and Black, M. J. (2015). Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 33

[Aliakbarian et al., 2020] Aliakbarian, S., Saleh, F. S., Salzmann, M., Petersson, L., and Gould, S. (2020). A stochastic conditioning scheme for diverse human motion prediction. In *CVPR*. 28

[Andriluka et al., 2014] Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 22, 56

[Ba et al., 2016] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*. 97

[Balakrishnan et al., 2018] Balakrishnan, G., Zhao, A., Dalca, A. V., Durand, F., and Guttag, J. V. (2018). Synthesizing images of humans in unseen poses. *CVPR*. 32

[Bautembach et al., 2018] Bautembach, D., Oikonomidis, I., and Argyros, A. (2018). Filling the joints: Completion and recovery of incomplete 3d human poses. *Technologies*. 35

[Bayat et al., 2021] Bayat, A., Sekuboyina, A., Paetzold, J. C., Payer, C., Stern, D., Urschler, M., Kirschke, J. S., and Menze, B. H. (2021). Inferring the 3d standing spine posture from 2d radiographs. 16

[Benzine et al., 2020] Benzine, A., Chabot, F., Luvison, B., Pham, Q. C., and Achard, C. (2020). Pandanet : Anchor-based single-shot multi-person 3d pose estimation. *CVPR*. 21

[Benzine et al., 2019] Benzine, A., Luvison, B., Pham, Q. C., and Achard, C. (2019). Deep, robust and single shot 3d multi-person human pose estimation from monocular images. *ICIP*. 21

[Benzine et al., 2021] Benzine, A., Luvison, B., Pham, Q. C., and Achard, C. (2021). Single shot 3d multi-person human pose estimation in complex images. *Pattern Recognition*. 21

[Biswas et al., 2019] Biswas, S., Sinha, S., Gupta, K., and Bhowmick, B. (2019). Lifting 2d human pose to 3d : A weakly supervised approach. *IJCNN*. 23

[Blanz and Vetter, 1999] Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 34

[Bogo et al., 2016] Bogo, F., Kanazawa, A., Lassner, C., Gehler, P. V., Romero, J., and Black, M. J. (2016). Keep it SMPL: automatic estimation of 3d human pose and shape from a single image. *ECCV*. 24, 25

[Bregler and Malik, 1998] Bregler, C. and Malik, J. (1998). Tracking people with twists and exponential maps. *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*. 33

[Carissimi et al., 2018] Carissimi, N., Rota, P., Beyan, C., and Murino, V. (2018). Filling the gaps: Predicting missing joints of human poses using denoising autoencoders. In *ECCV Workshops*. 35

[Carreira et al., 2016] Carreira, J., Agrawal, P., Fragkiadaki, K., and Malik, J. (2016). Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742. 74

[Chen and Ramanan, 2017] Chen, C.-H. and Ramanan, D. (2017). 3d human pose estimation = 2d pose estimation + matching. In *CVPR*. 24

[Chen et al., 2021] Chen, S., Zhang, L., and Zou, B. (2021). Estimation of 3d human pose using prior knowledge. *CORR*. 23

[Chen et al., 2019] Chen, X., Song, J., and Hilliges, O. (2019). Unpaired pose guided human image generation. *CVPR*. 31

[Chen et al., 2017] Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., and Sun, J. (2017). Cascaded pyramid network for multi-person pose estimation. *CVPR*. 84, 85

[Chen et al., 2022] Chen, Z., Zhao, X., and Wan, X. (2022). Structural triangulation: A closed-form solution to constrained 3d human pose estimation. 25

[Cheng et al., 2018] Cheng, Y., Zhao, W., Liu, C., and Tomizuka, M. (2018). Human motion prediction using adaptable neural networks. *CoRR*. 29

[Cho et al., 2014] Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*. 20

[Choi et al., 2021] Choi, H., Moon, G., Chang, J. Y., and Lee, K. M. (2021). 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 (CVPR)*. 26

[Choutas et al., 2020] Choutas, V., Pavlakos, G., Bolkart, T., Tzionas, D., and Black, M. J. (2020). Monocular expressive body regression through body-driven attention. In *ECCV*. 34, 68

[Chow and Liu, 1968] Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*. 33

[cocodataset, 2016] cocodataset (2016). Coco. https://cocodataset.org/#keypoints-2020. 22

[Corona et al., 2020] Corona, E., Pumarola, A., Alenyà, G., and Moreno-Noguer, F. (2020). Context-aware human motion prediction. *CVPR*. 28

[Cvetković et al., 2023] Cvetković, M. M., Desai, R., de Winkel, K. N., Papaioannou, G., and Happee, R. (2023). Explaining human body responses in random vibration: Effect of motion direction, sitting posture, and anthropometry. 14

[Dang et al., 2021] Dang, L., Nie, Y., Long, C., Zhang, Q., and Li, G. (2021). MSR-GCN: multi-scale residual graph convolution networks for human motion prediction. *ICCV*. 108

[Ding and Yin, 2021] Ding, P. and Yin, J. (2021). Uncertainty-aware human motion prediction. *CoRR*. 29

[Dong et al., 2019] Dong, J., Jiang, W., Huang, Q., Bao, H., and Zhou, X. (2019). Fast and robust multi-person 3d pose estimation from multiple views. *CVPR*. 25

[Dosovitskiy et al., 2021] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*. 19

[Du et al., 2016] Du, Y., Wong, Y., Liu, Y., Han, F., Gui, Y., Wang, Z., Kankanhalli, M., and Geng, W. (2016). Marker-less 3d human motion capture with monocular image sequence and height-maps. In *ECCV*. 58

[Duan et al., 2022] Duan, Y., Lin, Y., Zou, Z., Yuan, Y., Qian, Z., and Zhang, B. (2022). A unified framework for real time motion completion. *Proceedings of the AAAI Conference on Artificial Intelligence*. 102

[Einfalt et al., 2022] Einfalt, M., Ludwig, K., and Lienhart, R. (2022). Uplift and upsample: Efficient 3d human pose estimation with uplifting transformers. 26

[Felzenszwalb and Huttenlocher, 2005] Felzenszwalb, P. and Huttenlocher, D. (2005). Pictorial structures for object recognition. *International journal of computer vision*. 22

[Gelaw and Hagos, 2022] Gelaw, T. A. and Hagos, M. T. (2022). Posture prediction for healthy sitting using a smart chair. *International Conference on Advances of Science and Technology*. 16

[Ghezelghieh et al., 2016] Ghezelghieh, M. F., Kasturi, R., and Sarkar, S. (2016). Learning camera viewpoint using CNN to improve 3d body pose estimation. *3D Vision*. 32, 58

[Gholami et al., 2021] Gholami, S., Lorenzini, M., Momi, E. D., and Ajoudani, A. (2021). Quantitative physical ergonomics assessment of teleoperation interfaces. *IEEE Transactions on Human-Machine Systems*. 14

[Gong et al., 2021] Gong, K., Zhang, J., and Feng, J. (2021). Poseaug: A differentiable pose augmentation framework for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 30

[Gopalakrishnan et al., 2018] Gopalakrishnan, A., Mali, A. A., Kifer, D., Giles, C. L., and II, A. G. O. (2018). A neural temporal model for human motion prediction. *CVPR*. 28

[Groueix et al., 2018] Groueix, T., Fisher, M., Kim, V. G., Russell, B., and Aubry, M. (2018). AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 35

[Guan et al., 2022] Guan, S., Xu, J., He, M. Z., Wang, Y., Ni, B., and Yang, X. (2022). Out-of-domain human mesh reconstruction via dynamic bilevel online adaptation. *T-PAMI*. 61

[Guo et al., 2022] Guo, W., Du, Y., Shen, X., Lepetit, V., Alameda-Pineda, X., and Moreno-Noguer, F. (2022). Back to mlp: A simple baseline for human motion prediction. 27, 28

[Guo et al., 2023] Guo, W., Du, Y., Shen, X., Lepetit, V., Alameda-Pineda, X., and Moreno-Noguer, F. (2023). Back to mlp: A simple baseline for human motion prediction. 97, 99, 101, 102, 103, 104, 105, 108

[Hardy et al., 2022] Hardy, P., Dasmahapatra, S., and Kim, H. (2022). Optimising 2d pose representation: Improve accuracy, stability and generalisability within unsupervised 2d-3d human pose estimation. 23

[He et al., 2022a] He, C., Saito, J., Zachary, J., Rushmeier, H., and Zhou, Y. (2022a). Nemf: Neural motion fields for kinematic animation. In *NeurIPS*. 35

[He et al., 2022b] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022b). Masked autoencoders are scalable vision learners. In *CVPR*. xii, 73

[He et al., 2021] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. B. (2021). Masked autoencoders are scalable vision learners. *CVPR*. 101, 102, 103, 104, 105

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CVPR*. 18, 84, 85

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *NeurIPS*. xii, 74

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computing*. 20

[Huang et al., 2017] Huang, G., Liu, Z., and Weinberger, K. Q. (2017). Densely connected convolutional networks. *CVPR*. 19

[Huang et al., 2020] Huang, J., Zhu, Z., Huang, G., and Du, D. (2020). How to train your robust human pose estimator: Pay attention to the constraint cue. *CoRR*. 29

[Huang et al., 2022] Huang, L., Liang, J., and Deng, W. (2022). Dh-aug: Dh forward kinematics model driven augmentation for 3d human pose estimation. 31

[Hukkelås and Lindseth, 2023] Hukkelås, H. and Lindseth, F. (2023). Synthesizing anyone, anywhere, in any pose. 31

[Insafutdinov et al., 2016] Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schiele, B. (2016). Deepercut: A deeper, stronger, and faster multi-person pose estimation model. *ECCV*. 22

[Ionescu et al., 2014a] Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014a). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.* 56, 68, 69, 86

[Ionescu et al., 2014b] Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014b). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE TPAMI*. 99, 102, 103, 104

[Iqbal et al., 2020] Iqbal, U., Molchanov, P., and Kautz, J. (2020). Weakly-supervised 3d human pose learning via multi-view images in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 25, 58

[Jiang et al., 2023] Jiang, B., Chen, X., Liu, W., Yu, J., Yu, G., and Chen, T. (2023). Motiongpt: Human motion as a foreign language. *NeurIPS*. 115

[Jiang et al., 2022] Jiang, W., Jin, S., Liu, W., Qian, C., Luo, P., and Liu, S. (2022). Posetrans: A simple yet effective pose transformation augmentation for human pose estimation. 30

[Jin et al., 2020] Jin, S., Xu, L., Xu, J., Wang, C., Liu, W., Qian, C., Ouyang, W., and Luo, P. (2020). Whole-body human pose estimation in the wild. In *ECCV*. xi, 34, 68, 69, 85

[Johnson and Everingham, 2010] Johnson, S. and Everingham, M. (2010). Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*. 68

[Joo et al., 2015] Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., and Sheikh, Y. (2015). Panoptic studio: A massively multiview system for social motion capture. In *ICCV*. 68

[Joo et al., 2018] Joo, H., Simon, T., and Sheikh, Y. (2018). Total capture: A 3d deformation model for tracking faces, hands, and bodies. *CVPR*. 34, 68

[Ju et al., 1996] Ju, S., Black, M., and Yacoob, Y. (1996). Cardboard people: A parameterized model of articulated image motion. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*. 22

[Kadri et al., 2016] Kadri, H., Duflos, E., Preux, P., Canu, S., Rakotomamonjy, A., and Audiffren, J. (2016). Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17(1):613–666. 107

[Kanazawa et al., 2019] Kanazawa, A., Zhang, J. Y., Felsen, P., and Malik, J. (2019). Learning 3d human dynamics from video. In *CVPR*. 26

[Kang et al., 2023] Kang, Y., Liu, Y., Yao, A., Wang, S., and Wu, E. (2023). 3d human pose lifting with grid convolution. 24

[Katircioglu et al., 2021] Katircioglu, I., Georgantas, C., Salzmann, M., and Fua, P. (2021). Dyadic human motion prediction. *CoRR*. 28

[Kiciroglu et al., 2020] Kiciroglu, S., Wang, W., Salzmann, M., and Fua, P. (2020). Long term motion prediction using keyposes. *3DV*. 28

[Kissos et al., 2020] Kissos, I., Fritz, L., Goldman, M., Meir, O., Oks, E., and Kliger, M. (2020). Beyond weak perspective for monocular 3d human pose estimation. *CoRR*. 24

[Kolotouros et al., 2019] Kolotouros, N., Pavlakos, G., Black, M., and Daniilidis, K. (2019). Learning to reconstruct 3d human pose and shape via model-fitting in the loop. *ICCV*. 24

[Kreiss et al., 2021] Kreiss, S., Bertoni, L., and Alahi, A. (2021). OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association. *IEEE Transactions on Intelligent Transportation Systems*. vii, xi, 38, 68, 70, 90

[Krishna et al., 2021] Krishna, S., Vignesh, V., and Babu, D. (2021). Signpose -sign-language-animation-through-3d-pose-lifting.pdf. *ICCV*. 24

[Kuehne and Woerner, 2010] Kuehne, H. and Woerner, A. (2010). Motion segmentation of articulated structures by integration of visula perception criteria. *VisApp*. 22

[Lab, 2001] Lab, C. G. (2001). Motion capture database. *http://mocap.cs.cmu.edu*. 69, 80, 86

[Lehrmann et al., 2013] Lehrmann, A. M., Gehler, P. V., and Nowozin, S. (2013). A non-parametric bayesian network prior of human pose. In *2013 IEEE International Conference on Computer Vision*. 33

[Li et al., 2018] Li, C., Zhang, Z., Lee, W. S., and Lee, G. H. (2018). Convolutional sequence to sequence model for human dynamics. *CVPR*. 108

[Li et al., 2020] Li, S., Ke, L., Pratama, K., Tai, Y.-W., Tang, C.-K., and Cheng, K.-T. (2020). Cascaded deep monocular 3d human pose estimation with evolutionary training data. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 30, 58

[Li et al., 2022] Li, Z., Liu, J., Zhang, Z., Xu, S., and Yan, Y. (2022). Cliff: Carrying location information in full frames into human pose and shape estimation. *ECCV*. 61

[Li et al., 2019] Li, Z., Wang, X., Wang, F., and Jiang, P. (2019). On boosting single-frame 3d human pose estimation via monocular videos. In *ICCV*. 26

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *ECCV*. 22, 56

[Loper et al., 2015a] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015a). SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*. 24, 61

[Loper et al., 2015b] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015b). SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*. 99

[Lorenzini et al., 2021] Lorenzini, M., Kim, W., and Ajoudani, A. (2021). An online multi-index approach to human ergonomics assessment in the workplace. *IEEE Transactions on Human-Machine System*. 15

[Lutz et al., 2022] Lutz, S., Blythman, R., Ghosal, K., Moynihan, M., Simms, C., and Smolic, A. (2022). Jointformer: Single-frame lifting transformer with error prediction and refinement for 3d human pose estimation. *ArXiv*. 82, 83, 84, 85

[Luvizon et al., 2020] Luvizon, D. C., Tabia, H., and Picard, D. (2020). Multi-task deep learning for real-time 3d human pose estimation and action recognition. *TPAMI*. 26

[Luvizon et al., 2022] Luvizon, D. C., Tabia, H., and Picard, D. (2022). Consensus-based optimization for 3d human pose estimation in camera coordinates. *IJCV*. 21

[Ma et al., 2017] Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., and Gool, L. V. (2017). Pose guided person image generation. *NIPS*. 32

[Ma et al., 2011] Ma, R., Chablat, D., Bennis, F., and Ma, L. (2011). A framework of motion capture system based human behaviours simulation for ergonomic analysis. *14th International Conference on Human-Computer Interaction*. 15

[Mahmood et al., 2019] Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., and Black, M. J. (2019). AMASS: Archive of motion capture as surface shapes. In *ICCV*, pages 5442–5451. 102, 103, 104

[Mao et al., 2020] Mao, W., Liu, M., and Salzmann, M. (2020). History repeats itself: Human motion prediction via motion attention. *ECCV*. 99, 101, 102, 103, 104, 105, 108

[Mao et al., 2019a] Mao, W., Liu, M., Salzmann, M., and Li, H. (2019a). Learning trajectory dependencies for human motion prediction. *ICCV*. 28

[Mao et al., 2019b] Mao, W., Liu, M., Salzmann, M., and Li, H. (2019b). Learning trajectory dependencies for human motion prediction. *ICCV*. 108

[Marcon et al., 2017] Marcon, M., Pispero, A., Pignatelli, N., Lodi, G., and Tubaro, S. (2017). Postural assessment in dentistry based on multiple markers tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*. 16

[Martinez et al., 2017a] Martinez, J., Black, M. J., and Romero, J. (2017a). On human motion prediction using recurrent neural networks. In *CVPR*. 28

[Martinez et al., 2017b] Martinez, J., Black, M. J., and Romero, J. (2017b). On human motion prediction using recurrent neural networks. *CVPR*. 108

[Martinez et al., 2017c] Martinez, J., Hossain, R., Romero, J., and Little, J. J. (2017c). A simple yet effective baseline for 3d human pose estimation. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*. 23, 81, 82, 83, 84, 85

[Mehrizi et al., 2018] Mehrizi, R., Peng, X., Tang, Z., Xu, X., Metaxas, D. N., and Li, K. (2018). Toward marker-free 3d pose estimation in lifting: A deep multi-view solution. *FG2018*. 21

[Mehta et al., 2017] Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., and Theobalt, C. (2017). Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. 23, 56, 69, 80, 86

[Mejean, 2020] Mejean, V. (2020). Les operateurs n'ont peut-etre pas raison mais ils ont leurs raisons. http://www.ergonova.fr/

Les-operateurs-n-ont-peut-etre-pas-raison-mais-ils-ont-leurs-raisons. 10

[Mildenhall et al., 2020] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*. 35

[Mitchell, 1997] Mitchell, T. (1997). *Machine Learning*. McGraw-Hill. 16

[Mitra et al., 2020] Mitra, R., Gundavarapu, N. B., Sharma, A., and Jain, A. (2020). Multiview-consistent semi-supervised learning for 3d human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 58

[MMPose, 2020] MMPose (2020). Body keypoints. https://mmpose.readthedocs.io/en/latest/dataset_zoo/2d_body_keypoint.html,https://mmpose.readthedocs.io/en/latest/dataset_zoo/3d_body_keypoint.html. 41

[Moon et al., 2020] Moon, G., Yu, S., Wen, H., Shiratori, T., and Lee, K. M. (2020). Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single RGB image. *ECCV*. 68

[Moreno-Noguer, 2016] Moreno-Noguer, F. (2016). 3d human pose estimation from a single image via distance matrix regression. *CVPR*. 24

[Mudiyanselage et al., 2021] Mudiyanselage, S. E., Nguyen, P. H. D., Rajabi, M. S., and Akhavian, R. (2021). Automated workers ergonomic risk assessment in manual material handling using semg wearable sensors and machine learning. *Electronics*. 15

[Naeem et al., 2020] Naeem, M. F., Oh, S. J., Uh, Y., Choi, Y., and Yoo, J. (2020). Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*. 59

[Newell et al., 2016] Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. *ECCV*. 84, 85

[Noghre et al., 2022] Noghre, G. A., Pazho, A. D., Sanchez, J., Hewitt, N., Neff, C., and Tabkhi, H. (2022). Adg-pose: Automated dataset generation for real-world human pose estimation. *International Conference on Pattern Recognition and Artificial Intelligence*. 29

[Olivas-Padilla et al., 2022] Olivas-Padilla, B. E., Papanagiotou, D., Senteri, G., Manitsaris, S., and Glushkova, A. (2022). Computational ergonomics for task delegation in human-robot collaboration: spatiotemporal adaptation of the robot to the human through contactless gesture recognition. 16

[paperswithcode, 2020] paperswithcode (2020). Pose estimation. https://paperswithcode.com/task/pose-estimation. 10

[Park and Kwak, 2018] Park, S. and Kwak, N. (2018). 3d human pose estimation with relational networks. *BMVC*. 23

[Pavlakos et al., 2019] Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A., Tzionas, D., and Black, M. (2019). Expressive body capture: 3D hands, face, and body from a single image. *CVPR*. 34, 82

[Pavlakos et al., 2017] Pavlakos, G., Zhou, X., Derpanis, K. G., and Daniilidis, K. (2017). Coarse-to-fine volumetric prediction for single-image 3d human pose. *CVPR*. 21

[Pishchulin et al., 2016] Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V., and Schiele, B. (2016). Deepcut: Joint subset partition and labeling for multi person pose estimation. *CVPR*. 22

[Pumarola et al., 2019] Pumarola, A., Sanchez, J., Choi, G., Sanfeliu, A., and Moreno-Noguer, F. (2019). 3DPeople: Modeling the Geometry of Dressed Humans. In *International Conference in Computer Vision (ICCV)*. 32

[Qiu et al., 2019] Qiu, H., Wang, C., Wang, J., Wang, N., and Zeng, W. (2019). Cross view fusion for 3d human pose estimation. *ICCV*. 25

[Rhodin et al., 2018] Rhodin, H., Spörri, J., Katircioglu, I., Constantin, V., Meyer, F., Müller, E., Salzmann, M., and Fua, P. (2018). Learning monocular 3d human

pose estimation from multi-view images. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 21

[Rochette et al., 2021] Rochette, G., Russell, C., and Bowden, R. (2021). Human pose manipulation and novel view synthesis using differentiable rendering. *Face and Gesture*. 32

[Rong et al., 2021] Rong, Y., Shiratori, T., and Joo, H. (2021). Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. *ICCV*. 34

[Roy et al., 2022] Roy, P., Ghosh, S., Bhattacharya, S., Pal, U., and Blumenstein, M. (2022). Tips: Text-induced pose synthesis. 31

[Schwarcz and Pollard, 2019] Schwarcz, S. and Pollard, T. (2019). 3d human pose estimation from deep multi-view 2d pose. *ICPR*. 25

[Sengupta et al., 2021] Sengupta, A., Budvytis, I., and Cipolla, R. (2021). Probabilistic 3d human shape and pose estimation from multiple unconstrained images in the wild. *CVPR*. 21

[Shafti et al., 2018] Shafti, A., Ataka, A., Lazpita, B. U., Shiva, A., Wurdemann, H. A., and Althoefer, K. (2018). Real-time robot-assisted ergonomics. *ICRA*. 14

[Shotton et al., 2011] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *CVPR 2011*. 29

[Sidenbladh et al., 2000] Sidenbladh, H., la Torre, F. D., and Black, M. (2000). A framework for modeling the appearance of 3d articulated figures. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*. 22

[Sigal et al., 2011] Sigal, L., Isard, M., Haussecker, H., and Black, M. J. (2011). Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision*. 33

[Simonyan and Zisserman, 2015] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *ICLR*. 18

[Skorvánková et al., 2021] Skorvánková, D., Riecický, A., and Madaras, M. (2021). Automatic estimation of anthropometric human body measurements. *CVPR*. 16

[Sminchisescu et al., 2006] Sminchisescu, C., Kanaujia, A., and Metaxas, D. (2006). Learning joint top-down and bottom-up processes for 3d visual inference. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. 32

[Sun and Chowdhary, 2023] Sun, J. and Chowdhary, G. (2023). Towards accurate human motion prediction via iterative refinement. 28

[Sun et al., 2021] Sun, J., Lin, Z., Han, X., Hu, J.-F., Xu, J., and Zheng, W.-S. (2021). Action-guided 3d human motion prediction. In *NeurIPS*. 28

[Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. *CVPR*. 18

[Takahashi et al., 2018] Takahashi, K., Mikami, D., Isogawa, M., and Kimata, H. (2018). Human pose as calibration pattern; 3d human pose estimation with multiple unsynchronized and uncalibrated cameras. In *CVPR Workshops*. 26

[Tang et al., 2021] Tang, J., Yuan, Y., Shao, T., Liu, Y., Wang, M., and Zhou, K. (2021). Structure-aware person image generation with pose decomposition and semantic correlation. *CoRR*. 32

[Tanke et al., 2019] Tanke, J., Weber, A., and Gall, J. (2019). Human motion anticipation with symbolic label. *CoRR*. 28

[Tekin et al., 2015] Tekin, B., Rozantsev, A., Lepetit, V., and Fua, P. (2015). Direct prediction of 3d body poses from motion compensated sequences. *CVPR*. 26

[Tevet et al., 2023] Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., and Bermano, A. H. (2023). Human motion diffusion model. *CVPR*. 115

[Trumble et al., 2017] Trumble, M., Gilbert, A., Malleson, C., Hilton, A., and Collomosse, J. (2017). Total capture: 3d human pose estimation fusing video and inertial sensors. In *BMVC*. 99

[Varol et al., 2021] Varol, G., Laptev, I., Schmid, C., and Zisserman, A. (2021). Synthetic humans for action recognition from unseen viewpoints. *IJCV*. 32

[Varol et al., 2017] Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. (2017). Learning from synthetic humans. In *CVPR*. 32, 58

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *NIPS*. xi, 19, 73

[von Marcard et al., 2018a] von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., and Pons-Moll, G. (2018a). Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*. 99, 102, 103, 104

[von Marcard et al., 2018b] von Marcard, T., Henschel, R., Black, M. J., Rosenhahn, B., and Pons-Moll, G. (2018b). Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 61, 68

[von Marcard et al., 2018c] von Marcard, T., Henschel, R., Black, M. J., Rosenhahn, B., and Pons-Moll, G. (2018c). Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*. 108

[Wandt et al., 2021] Wandt, B., Rudolph, M., Zell, P., Rhodin, H., and Rosenhahn, B. (2021). Canonpose: Self-supervised monocular 3d human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 24, 53, 57, 58, 61, 81, 82, 83, 84

[Wang et al., 2020] Wang, J., Yan, S., Xiong, Y., and Lin, D. (2020). Motion guided 3d pose estimation from videos. In *ECCV*. 26

[Wei et al., 2019] Wei, G., Lan, C., Zeng, W., and Chen, Z. (2019). View invariant 3d human pose estimation. *CoRR*. 23

[Weinzaepfel et al., 2020] Weinzaepfel, P., Brégier, R., Combaluzier, H., Leroy, V., and Rogez, G. (2020). DOPE: distillation of part experts for whole-body 3d pose estimation in the wild. *ECCV*. 34, 85

[Wenzheng et al., 2016] Wenzheng, C., Wang, H., Li, Y., Su, H., Tu, C., Lischinski, D., Cohen-Or, D., and Chen, B. (2016). Synthesizing training images for boosting human 3d pose estimation. *3D Vision*. 31

[WHO, 2022] WHO (2022). musculoskeletal-conditions. https://www.who.int/news-room/fact-sheets/detail/musculoskeletal-conditions. 15

[wikipedia, 2001] wikipedia (2001). Spherical coordinate system. https://en.wikipedia.org/wiki/Spherical_coordinate_system. 47

[wikipedia, 2002] wikipedia (2002). Markov chain. https://en.wikipedia.org/wiki/Markov_chain. 48

[wikipedia, 2003a] wikipedia (2003a). Heat equation. https://en.wikipedia.org/wiki/Heat_equation. 52

[wikipedia, 2003b] wikipedia (2003b). Motion capture. https://en.wikipedia.org/wiki/Motion_capture. 11

[Wu et al., 2022] Wu, J., Wang, J., Si, S., Qu, X., and Xiao, J. (2022). Pose guided human image synthesis with partially decoupled gan. 32

[Xiang et al., 2019] Xiang, D., Joo, H., and Sheikh, Y. (2019). Monocular total capture: Posing face, body, and hands in the wild. In *CVPR*. 34, 68

[Xu et al., 2020] Xu, H., Bazavan, E. G., Zanfir, A., Freeman, W. T., Sukthankar, R., and Sminchisescu, C. (2020). Ghum amp; ghuml: Generative 3d human shape and articulated pose models. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 31

[Xu et al., 2021] Xu, J., Chen, X., Lan, X., and Zheng, N. (2021). Probabilistic human motion prediction via A bayesian neural network. *ICRA*. 29

[Xu and Takano, 2021] Xu, T. and Takano, W. (2021). Graph stacked hourglass networks for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 24

[Yang et al., 2018] Yang, C., Wang, Z., Zhu, X., Huang, C., Shi, J., and Lin, D. (2018). Pose guided human video generation. *ECCV*. 31

[Yasar and Iqbal, 2021] Yasar, M. S. and Iqbal, T. (2021). Improving human motion prediction through continual learning. *CoRR*. 28

[Yazdani et al., 2021a] Yazdani, A., Novin, R. S., Merryweather, A., and Hermans, T. (2021a). DULA: A differentiable ergonomics model for postural optimization in physical HRI. *CoRR*. 14

[Yazdani et al., 2021b] Yazdani, A., Novin, R. S., Merryweather, A., and Hermans, T. (2021b). Ergonomically intelligent physical human-robot interaction: Postural estimation, assessment, and optimization. *AAAI-FSS*. 14

[Yazdani et al., 2022] Yazdani, A., Novin, R. S., Merryweather, A., and Hermans, T. (2022). Dula and deba: Differentiable ergonomic risk models for postural assessment and optimization in ergonomically intelligent phri. 14

[Zell et al., 2017] Zell, P., Wandt, B., and Rosenhahn, B. (2017). Joint 3d human motion capture and physical analysis from monocular videos. In *CVPR Workshops*. 26

[Zhang et al., 2021] Zhang, C., Zhan, F., and Chang, Y. (2021). Deep monocular 3d human pose estimation via cascaded dimension-lifting. *CORR*. 24

[Zhang et al., 2020a] Zhang, J., Liu, X., and Li, K. (2020a). Human pose transfer by adaptive hierarchical deformation. *CoRR*. 32

[Zhang et al., 2020b] Zhang, Y., Briq, R., Tanke, J., and Gall, J. (2020b). Adversarial synthesis of human pose from text. *DAGM GCPR*. 31

[Zhou et al., 2016] Zhou, X., Zhu, M., Leonardos, S., Derpanis, K. G., and Daniilidis, K. (2016). Sparseness meets deepness: 3d human pose estimation from monocular video. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 27

[Zhu and Picard, 2022] Zhu, Y. and Picard, D. (2022). Decanus to legatus: Synthetic training for 2d-3d human pose lifting. *ACCV*. 12

[Zhu et al., 2023a] Zhu, Y., Rudenko, A., Kucner, T. P., Lilienthal, A. J., and Magnusson, M. (2023a). A data-efficient approach for long-term human motion prediction using maps of dynamics. 28

[Zhu et al., 2023b] Zhu, Y., Samet, N., and Picard, D. (2023b). H3wb: Human3.6m 3d wholebody dataset and benchmark. *ICCV*. 12

[Zhu et al., 2024] Zhu, Y., Samet, N., and Picard, D. (2024). Pius: Pose interpolation at extremely low and uneven framerate. *In submitting*. 12

[Zimmermann and Brox, 2017] Zimmermann, C. and Brox, T. (2017). Learning to estimate 3d hand pose from single RGB images. *ICCV*. 68

[Zimmermann et al., 2019] Zimmermann, C., Ceylan, D., Yang, J., Russell, B. C., Argus, M., and Brox, T. (2019). Freihand: A dataset for markerless capture of hand pose and shape from single RGB images. *ICCV*. 68