Video-Based Human-Posture Monitoring from RGB-D Cameras

Anonymous Authors

***Abstract*—Correct pose/posture is crucial in most human activities, and increasingly in using computer screens of many form factors. In this paper, we build a spatiotemporal reasoning infrastructure on top of standard Computer Vision (CV) algorithms to provide an alternate, much more accurate, faster method for tracking correct posture than pure deep learning (DL) methods. We use CV to determine poses of the 2D human stick models from RGB images, which are further enhanced using depth information (from RGB-D camera) to determine relevant angles and compare them against the standards. By applying our method to two very different posture applications (knowledge worker and taekwondo), we show that it outperforms all others, including machine learning, deep learning, and time series-based prediction. Furthermore, superior performance is seen not only in the estimation accuracy but also in the estimation speed.**

***Impact Statement*—Correct posture is critical from many perspectives including general health, avoidance of injuries, ensuring correct movements in various performing arts, aesthetics, etc. This paper demonstrates that how the integration of computer vision with logic reasoning can provide posture monitoring not only more accurately but also much faster than the direct deep learning techniques in vogue. The efficiency is important to enable the monitoring in real-time, as required by most applications. Such monitoring can enable feedback for posture correction for manually unsupervised activities (e.g., office work), and reduce manual effort in supervised activities (e.g., performing art classes.)**

***Index Terms*—Posture Recognition, Depth Correction, RGB-D cameras, 3D pose detection**

I. Introduction

Correct pose and pose transitions are crucial in numerous different contexts and may be necessary from many different perspectives, such as aesthetics (e.g., dancing), successful performance of an activity (e.g., gymnastics), avoiding severe injuries (e.g., lifting weights), avoiding repetitive injuries (e.g., assembly line work), and mitigating adverse effects of incorrect sitting, which has been called the "new smoking" [1]. In many cases, a correct pose transition is as important as the pose itself (e.g., dancing, martial arts, gymnastics, weight lifting, etc.) Note that in all these applications, the pose is not simply a qualitative concept (e.g., standing vs. sitting) but requires precise quantification in terms of 3D angles of various body parts (e.g., the angle between the upper arm and the main body). However, there are naturally some allowable ranges/tolerances.

This paper aims to demonstrate that we can accurately determine such 3D angles and body dimensions from RGB-D camera views, thereby enabling comparison of the pose and pose transition against the given standards or requirements. The uniqueness of our approach is to do so by building a spatio-temporal reasoning infrastructure on top of basic computer vision algorithms. Vision algorithms are limited to detecting objects (humans) and constructing their 2D stick models, and thus, they are accomplished without any additional or specialized training data. Using the reasoning avoids any need for additional training data and associated problems of bias or overfitting. We consider two distinct applications in this paper: (a) A knowledge worker sitting in front of a computer monitor, where we show that the video-based posture 3D angle estimations accurately track the elaborate ground truth measurements that we conducted using a host of sensors, and (b) A martial arts (Taekwondo) scenario, where the focus is only correctly determining pose transitions using logic reasoning. In both applications, we show that we can accurately estimate the depth dimension using a unique geometric approach and, from there, the 3D angles, poses, and pose transitions, all without any additional training data or neural net models. We compare our results against those of a variety of other methods in the literature, all of which use some form of neural nets (and hence require additional training data), and demonstrate that we not only achieve a higher accuracy but also faster running time of around 500 ms in *all* cases without exceptions.

The rest of the paper is organized as follows. Section II discusses the existing work-related posture standards. Section III discusses the related work and lays out our contributions. In section IV we describe the proposed 3D posture recognition approach using RGB-D cameras. Section V then describes our 2D pose transition framework. Section VI comprehensively evaluates the proposed frameworks. Finally, Section VII concludes the discussion.

II. Background

A. Posture Standards

Human posture in an ergonomic work environment has been studied extensively where it depends significantly on the furniture used, e.g., chair and table heights, contact with the chair backrest, screen height/distance, etc. Thus, the ergonomics standards defined in various countries such as the USA, Australia, Europe, and Japan [2] include both human and furniture aspects. Several methods have been developed for the assessment of body poses for various types of tasks, including REBA, RULA, WISHA, OWAS, etc. [3], [4], most of which are directed toward estimating the risk of Musculoskeletal Disorder (MSD). We review some of these in the following.

Changes in posture provide valuable information about stress and attention levels. Several gadgets have also been

developed for posture monitoring, such as a textile pressure sensor [5] or a censored smart cushion for the chair back [6]. There are also some proprietary tools to check worker posture using a smartphone, such as https://www.tumeke.io/, or https://www.ehs.com/2020/03/your-work-from-home-toolbox/, but it is unclear how they work.

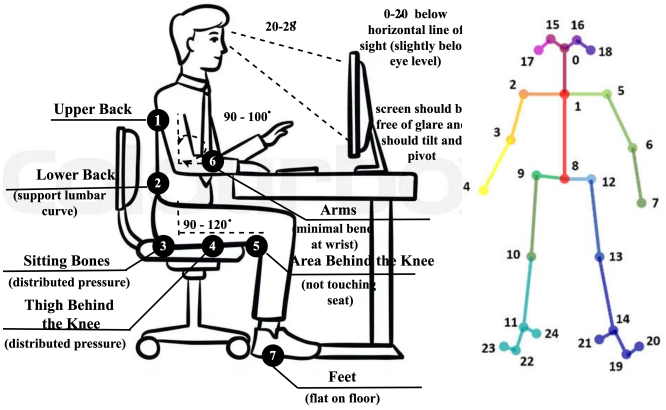


Fig. 1: (a) Posture recommendations, (b) Openpose Model

The posture standards are primarily designed for informal manual use and have many gaps in the requirements. In particular, while some angles are specified, several others are not, as shown in Fig 1(a). In these cases, we make some assumptions. Also, qualitative specifications like "back should be straight" do not provide tolerance. We assume a tolerance value of $\pm 10^{\circ}$ primarily based on our judgment of when the deviations become noticeable.

III. Related Work and Our Contributions

A. 2D Posture Modeling

Human pose estimation based on images is a well-developed area with several pose models and algorithms. 2D pose models assume a "stick" representation of humans defined by a set of "key points" and body segments (or parts) connecting those key points. YOLO-v8 is the latest of high successful YOLO series of of so-called "top-down" methods (i.e., recognize person first and then the body parts). The survey paper [7] provides a comprehensive overview of popular top-down approaches for human pose estimation. A prevalent bottom-up method (i.e., that assembles person from the body parts) is OpenPose [8]. Sitting posture recognition based on OpenPose is discussed in [9]. We used OpenPose in this research, although YOLO could also have been used.

OpenPose only concerns the 2D image (i.e., no depth). Furthermore, the body segments are merely sticks without any width information.[1] Thus, further analysis of the 2D image also needs to determine the width. These 2D models vary; for example, COCO (or body_17) has no keypoints for hands/feet, but the body_25 human model shown in Fig. 1(b) does. Another model called Blazepose [10] is even more detailed than body_25. In this paper, we use body_25 as it is detailed enough for most posture studies; one could similarly use others for applications such as dance, where the precise orientation of fingers and toes may be necessary. It is worth noting that although accurate 3D (human) body/pose models (as opposed to stick models) do exist [11], [12], they are not usable for real-time recognition and are deeply affected by clothing.

[1]Openpose assigns a width to body segments, but these are predetermined hyperparameters.

B. 3D Posture Recognition from RGB-Depth Cameras

Determining body key points in 3D space is critical to action recognition, human-robot interaction, and sports analysis. However, this information is challenging to acquire using standard RGB cameras, despite considerable efforts to estimate depth from plain RGB frames or "monocular" images [13]. There are also some recent approaches [14], [15] that use scale-cues extracted from 2D images to attempt to predict absolute 3D poses; the results are merely approximations. Such methods need to be both accurate and fan enough for real-time use.

To obtain a good depth measure in real-time, having multiple views of the scene from different angles is essential. This can be done by using various RGB cameras deployed at a known distance apart or by the emerging RGB-D cameras that are more easily deployed. Scanning LIDARs also provide depth information similar to RGB-D cameras but are much more expensive. In this paper, we only work with RGB-D cameras. Authors in [16] propose a method to segment body parts using a random forest classifier to perform per-pixel classification of parts and cluster these pixels for each part to localize 3D key points using single depth images. Ref [17] develops an approach for human pose estimation from depth images by building upon Hough forests [18]. Further, [19] transforms the depth map into a voxelized grid and performs a voxel-to-voxel prediction using a 3D CNN. Such methods suffer from high computational complexity, which was further addressed in [20] where an anchor-based method applies 2D convolution directly to depth images. Reference [21] evaluates the relative accuracy of various 3D pose estimation methods.

We aim to use existing 2D pose estimation methods by incorporating our distinct depth enhancement technique to extract precise 3D information regarding the positions of body joints and the angles of body segments. In this paper, we take a unique approach to depth estimation that exploits various geometric constraints and works directly on the data without having to train a model or using complex calculations.

IV. Proposed System for 3D Posture Recognition

The proposed active posture monitoring system obtains 3D joint locations and joint angles from an RGB-D camera (Intel® RealSense™ D435i) output by combining OpenPose (on RGB image) and a unique way of processing the depth map. For comparison, we obtain the anatomical 3D body landmark positions using Xsens MTw™, a miniature wireless inertial measurement unit incorporating 3D accelerometers, gyroscopes, magnetometers (3D compass), and a barometer (pressure sensor), as discussed in the section VI-A.

Fig. 2 illustrates the stages involved in our 3D posture recognition approach. We start with the skeleton recognition using the body-25 OpenPose model and then pick 11 essential

key points of interest: neck, nose, right/left shoulder, right/left wrist, right/left elbow, right/left, and mid-hip joint locations representing the upper human body. Any two joints form one skeletal segment. For instance, the neck and nose key points, denoted by 0 and 1, comprise a skeletal segment. Once joint positions are extracted using OpenPose, we compute the geometrical angles of the adjacent segments. The steps involved in the proposed 3D posture recognition are shown in Algorithm. 1. The proposed algorithm primarily comprises four steps: (1) Extracting 2D keypoints and joint angles using OpenPose, (2) Denoising the depth map and calculating new depth values from the comprehensive depth map, taking into account the arms/limbs between each detected keypoint pair, (3) Estimating 3D keypoints and 3D joint angles, and (4) Classifying the posture. The time complexity of the proposed algorithm is primarily dependent upon the entries in the global arrays L and W, resulting in $O(LW)$.

Algorithm 1 Human-Posture Monitoring Algorithm

```
void Posture Recognition_Algorithm() //
Step 1: Detect 2D Keypoints and Joint angles
    if person == True:
        extract((x_a,y_a),(x_b,y_b), ...) using Openpose //2D Keypoints
        extract((θ_ab),θ_bc) ...) using Openpose //2D Joint angles
    append(2D Keypoints and Joint Angles vectors)
Step 2: Depth Transformation
    L ← Global array of all the lengths between keypoint pairs
    W ← Global array of widths of all limbs and arms between keypoint pairs
        for each entry (l,w) in L and W:
            estimate new depth value, d'(l,w) using Eqn. 1
        append(Denoised 3D Depth Map)
Step 3: Detect 3D Keypoints and 3D Joint angles
    if person == True:
        extract((x_a,y_a,z_a),(x_b,y_b,z_b), ...) using Eqn. 3 //3D Keypoints
        extract((φ_ab),φ_bc) ...) using Eqn. 4 //3D Joint angles
    append(3D Keypoints and Joint Angles vectors)
Step 4: Determine Posture or Posture Transitions
    Classify Posture using SMT assertions as shown in Example 1
```

A. 2D Keypoints Extraction and Image Alignment

The OpenPose deep learning network uses two subnets each with $K=6$ *stages* [22]. One branch estimates the *Confidence Map* (CM), which gives the key point locations of each object. The second branch estimates the *Part Affinity Field* (PAF), which offers all limb orientations. Thus, OpenPose has the keypoint CMs of all the agents and limb orientations as PAFs during inferencing, providing the XY angles. Because of its bottom-up nature, the association between limbs and people present in the scene is done via the Hungarian algorithm to ensure that each keypoint κ_1 is connected to at most one keypoint κ_2 as specified by the pose model. As for the width, we can estimate it by combining domain knowledge (e.g., range of arm widths) and texture changes in the direction perpendicular to the limb directions. Since free-flowing clothing can make this very difficult, we will assume throughout that the clothing largely conforms to the body shape.

Next, we use the depth cloud from the RGB-D camera to estimate the depths and, hence, the XZ and YZ angles. For this, we first align the RGB and depth images so that the two overlap and are well-aligned. The depth estimation mechanism results in a different focal length and offset than the RGB camera; thus, the RGB image and depth map do not overlap completely. The depth map viewed as an image is of poor quality (see the 4th panel in Fig. 2); therefore, we cannot reliably run open-pose on it and extract corresponding key points. Thus, directly matching key points between RGB image and depth map is impossible. Fortunately, the discrepancy between the two is relatively stable across frames, making it easy to align them with a simple transformation. This allows us to describe each image pixel using its (x,y,z) coordinate. Since the camera position and parameters are known, we also apply the inverse projection transformation to transform the image coordinates into the object coordinates.

B. Depth Transformation

Depth maps captured from RGB-D cameras contain many defects in the form of both missing depth values and incorrect ones, largely due to scattering of light (including reflection, diffraction, etc.). The defects may vary from frame to frame, particularly when there is movement. Thus, we can simply use the estimated z values of each pixel and instead need to do some processing to remove or minimize artifacts. Many approaches exist, ranging from simple filter-based methods to learning-based methods that use deep neural networks to enhance the depth maps (see, for example, the survey paper [23]). Numerous methods have been investigated to curate the depth map and integrate it with RGB image, which involves some form of deep learning [24]–[26].

Instead of following a traditional approach to reducing noise in-depth maps, we apply a novel geometric approach to improve the quality of depth estimation for the 3D pose of the people. A similar mechanism can be used for other significant objects that we want to model in detail. We estimate the depth limb-wise guided by the pose model. Let's take an example of the upper right arm, characterized by key points 2 and 3 as estimated by OpenPose. We will use simple geometric arguments to calculate the depth and, hence, the angle by assuming that the arm is roughly straight and in the form of a cylinder. Let L denote the length (distance between key points 2 and 3) and W the arm's radius (or half-width). In posture applications, a precise estimate of W is not needed; therefore, we estimate it simply using the procedure below.

To estimate W, we choose a set of equidistant points between the keypoints of an arm segment and determine width based on the maximal change in the color intensity while limiting it to the maximum reasonable value under our assumption of the clothing mainly conforming to the body. We then take the median width as the actual width to discount the impact of overlaps assumed to be infrequent. We have observed that the accuracy of determining the width is close to 92% compared to the ground truth of the width of different body segments.

Let θ be the depth gradient to be estimated, and $d(l,w)$ the measured depth at a point **(pixel)** at a distance $l \in [0..L]$ from keypoint two and a distance $-W \leq w \leq W$ from the center of the arm boundary. Now, if the arm were a perfect cylinder and the depth measures did not have any noise, we can map the

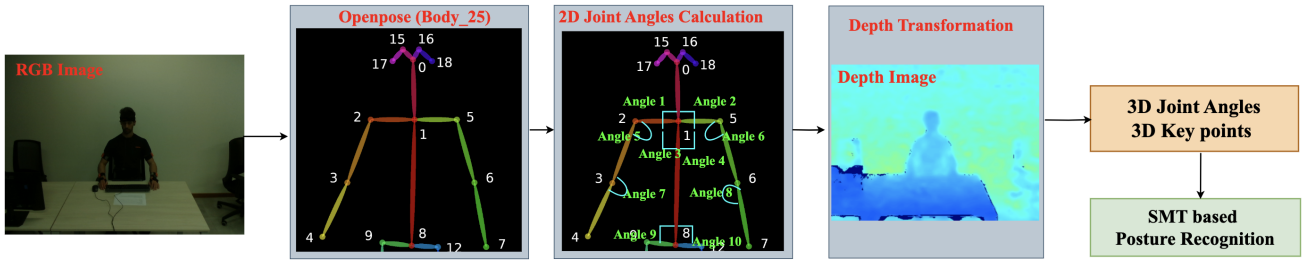


Fig. 2: 3D Posture Recognition from RGB-D cameras

depth at every point in the arm to a single point by using the following transformation:

$$d'(l,w) = d(l,w) - l\sin\theta - W + \sqrt{W^2 - w^2} \quad (1)$$

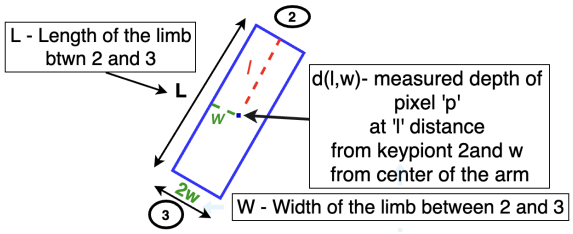


Fig. 3: Estimation and Curation of Depth

That is, $d'(l,w) = C$, where C is a constant independent of l and w (equal to the true depth at Keypoint 2). Since the arm is not a perfect cylinder and the pixel-in-depth maps are noisy, we expect $d'(l,w)$ values to be scattered but clustered around the point C. We thus remove the outliers and find the cluster's center as the true depth. Practically, this requires finding the cluster's center using all the data, removing points falling outside the boundary, and then repeating the process until no outliers are found.

Since θ is unknown in the equation above, we can first estimate C, say C_0, by considering only the points with $l = 0$. Next, we choose $l = L$ and compute:

$$\sin\theta = [d(L,w) - C_0 - W + \sqrt{(W^2 - w^2)}]/L \quad (2)$$

These values will form a cluster, and we successively remove the outliers and ultimately get an initial estimate of θ, say θ_0. We can now set up an iterative procedure to estimate C_n and θ_n from C_{n-1} and θ_{n-1} until convergence is achieved.

The procedure will be repeated for successive limbs, starting with the known depth at the previous key points. For example, when computing the angle for limb (2,1), we begin with the depth at keypoint 2, which is already known. This successive procedure ensures joint consistency, although more sophisticated procedures can also be used.

Following the depth estimation, we have the 3D vectors for each limb. For example, the vector for a limb from keypoint a to keypoint b is given by:

$$v_i(v_x, v_y, v_z) = (x_b - x_a, y_b - y_a, y_b - y_a) \quad (3)$$

Let ϕ_{ij} denote the 3D angle between two relevant limbs (e.g., upper and lower arms, upper arm and back, etc.) represented by vectors v_i and v_j. Then ϕ_{ij} is given by the dot-product divided by the product of magnitudes, i.e.,

$$\phi = (v_i \cdot v_j)/(|v_i| \times |v_j|) \quad (4)$$

C. Posture and Posture Transition Recognition

Posture recognition can be considered a multi-class classification problem, and the prevailing methods would train a machine learning model for the classification. For example, in the context of the posture of a knowledge worker, we have identified five relevant poses as explained in Section VI-A, such as crouching, slouching, etc. One could either modify pose detection algorithms or build additional layers on top of feature extraction neural net such as VGG19 to identify these postures. However, we do not follow such a path; instead, we show that we can do substantially better without requiring any training data using an explicit logical reasoning method. We also use the same method to recognize pose transitions that are needed for detecting if activities requiring specific body movements are done correctly (e.g., martial arts, dance, etc.) Pose transition recognition using a direct machine learning model is awkward at best.

To address pose classification, we note that each pose can be characterized in terms of 3D angles of various body parts. The relevant angles for our knowledge worker dataset are depicted in Fig 2 and will be discussed later. We formulate the classification/recognition problem using first-order logic, i.e., as a Boolean satisfiability problem in terms of the well-known Satisfiability Modulo Theories (SMT) [27]. The logic formulae obviously need to include a range of acceptable angles, which the SMT can handle easily by including relational algebra and arithmetic. SMT is a well-developed technology that we have used extensively in the past and works extremely fast and well, despite theoretical issues like the undecidability of general first-order logic or the NP-hardness of boolean satisfiability problems.

For example, consider the slouching posture, characterized by a person sitting with rounded shoulders, a forward-leaning head, and a slightly bent neck. Therefore, the slouching posture can be described as the combination of angles 1 and 3, which indicate a slight forward bending of the head, along with angles 5, angle 6, and angle 9, which represent the angles at which the shoulders are bent and hanging. If any of the following combinations of angles are outside the desired range of values - angle 1, angle 3, or 5, 6, and 9 - the posture is classified as "slouching" using SMT assertions. The following statement, which is given as input to Z3 [28], a popular SMT tool, demonstrates the recognition of slouching:

The assertions in Definition. 1 and 2 consists of five blocks. The first line selects the underlying theory, QF_LIA (linear integer arithmetic). The following lines are used to declare variables or functions. Variables are declared using (declare-

(set-logic QF_LIA)
(declare-const angle_1 Int), *(declare-const angle_3 Int)*
(declare-const angle_5 Int), *(declare-const angle_6 Int)*
(declare-const angle_9 Int)
(assert (and (or (> angle_1 15) (> angle_3 90))
(or (≥ angle_9 90) (> angle_3 90)) (or (> angle_1 15 (≤ angle_5 value) (≥ angle_6 value) (≥ angle_9 value))) (or (> angle_3 90) (≤ angle_5 value) (≥ angle_6 value) (≥ angle_9 value)))
(check-sat); *(get-model)*; (exit)

Definition 1: Assertion in SMT to classify slouching posture

const name type), where name is the variable name and type is the variable type. Constraints/rules/assertions in SMT-LIB are of the form (assert ...) describing the rules expressed in the Conjunctive Normal Form (CNF). The command (check-sat) solves the assertions defined using (assert ...). Depending on the result, one can then use (get-model) to obtain a model when the formula is satisfiable or (get-unsat-core) to extract an unsatisfiable core (a subset of the rules) when the formula is unsatisfiable. Finally, the command (exit) terminates the solver.

(set-logic QF_LIA)
(declare-const Direction)
(assert (= Direction "Forward","Hold"))
(declare-const L_Walking_Stance, R_Walking_Stance Int)
(declare-const L_Front_Kick, R_Front_Kick Int)
(declare-const L_Front_Stance , R_Front_Stance Int)
(declare-const L_Back_Stance , R_Back_Stance Int)
(assert (and (= L_Walking_Stance 90) (= R_Walking_Stance 180)
(assert (and (= R_Front_Kick, R_Front_Stance "Forward")
(assert (and (= R_Front_Kick, R_Front_Stance "Forward")
(assert (and (= L_Walking_Stance ,R_Walking_Stance 90)
(assert (and (= R_Back_Stance 90) (= L_Back_Stance 180)
(assert (and (= L_Front_Stance, R_Front_Stance "Hold")
(assert (and (= L_Walking_Stance 90) (= L_Walking_Stance 270)
(assert (and (= L_Front_Kick, L_Front_Stance "Forward")
(assert (and (= R_Walking_Stance 180) (= L_Walking_Stance 90)
(assert (and (= L_Front_Kick, L_Walking_Stance, R_Front_Kick, R_Walking_Stance "Forward")
(check-sat); *(get-model)*; (exit)

Definition 2: Assertion in SMT to classify pose into proper or improper in Green Belt Pattern

V. Recognizing Pose Transitions

Taekwondo is a martial art similar to gymnastics, which necessitates the precise execution of movements with proper techniques. The exact order for pose transitions for every belt pattern in Taekwondo is detailed in the source [29] and can be put into a lookup table.

Our proposed 2D pose transition framework identifies the correctness of each pose during pose transitions and monitors the proper transition of poses in Taekwondo. For 2D key-point joint locations extracted from RGB images, we utilize Openpose. In this approach to 2D pose transitions, 15 distinct key points are obtained and numbered from 0 to 14, while the remaining key points (15–24) are omitted. Equation.4 is utilized in a two-dimensional coordinate system to estimate the angle between any two skeletal segments. Subsequently, we retrieve the accurate sequence of pose transitions by maintaining a lookup table. Each pose within the transition sequence is evaluated for its correctness using the SMT tool, considering factors such as the angles of movement in various body parts like legs and arms. Taekwondo instructors cross-check and validate these angles to ensure accuracy and reliability.

Each pose is classified based on the angle values of the athlete's legs and arms. Let us examine a green belt movement pattern, waking stance low block, shown in Fig. 5. This pattern entails an athlete maintaining a straight stance while extending the right arm in a straight line while the other arm is positioned at the waist. When the athlete assumes this position while standing, exhibiting rounded shoulders, a forward-leaning head or back, and arm placement that deviates from the permitted range, it can be considered an improper pose. A sequence of poses that begins with such poor posture and continues with it is not beneficial in training for particular belt patterns. Hence, the waking stance low block constitutes an improper posture characterized by a combination of angles 1 and 3, signifying a marginal forward inclination of the head and the back, in addition to angles 5, 6, and 9, denoting the positions at which the shoulders bend. Additionally, the extension of the right arm in a straight position can be verified through angle 7 (these angles follow the same numbering as in Fig. 2). The posture is considered "improper" based on SMT assertions if any of the subsequent combinations of angles fall outside the intended range of values. The SMT assertions to classify the walking low stance as improper follow a similar syntax as in Definition 1. The SMT tool verifies the correctness of each posture during pose transitions, allowing instructors to provide feedback and correct the postures of their students while they train.

VI. Experimental Setup and Results

In this section, we evaluate our posture recognition framework on both of our collected datasets discussed below. The following metrics determine the effectiveness of our framework: (a) the accuracy and (b) the average execution time of posture recognition. The experiments were performed on a computer with Intel(R) Core(TM) i7-7700 CPU @ 3.60 GHz, 32 GB RAM, and 1 TB SSD.

A. Data collection

To evaluate the performance of the proposed method, we created a new RGB-D dataset using the Intel® RealSense™ D435i. In addition to RGB images and depth maps for each observation, the dataset includes the 3D joint coordinates using a second-generation wireless inertial-magnetic motion tracker by Xsens.

1) Posture Dataset: The data collection was designed to replicate an authentic office work environment, focusing on a worker engaged in computer-related tasks. We organized a desk, chair, keyboard, and mouse to emulate a typical workstation. The Intel® RealSense™ D435i camera was positioned precisely on the same axis as the chair to capture detailed images. This setup was maintained at 250 cm from the subject, elevated to a height of 185 cm, and oriented at an angle of -15 degrees relative to the ground, as shown in Fig. 4.

The participant was outfitted with the Xsens motion capture system, which comprises 17 strategically placed sensors on the subject's body, as also depicted in Fig. 4. The sensor locations were chosen to be at vital anatomical landmarks

Fig. 4: Experimental Setup

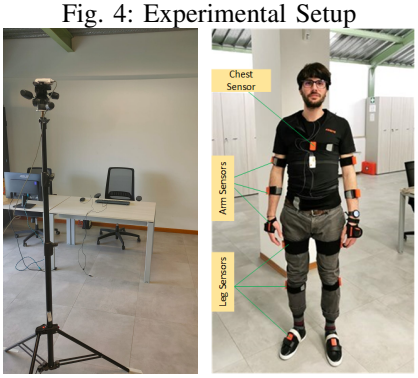


TABLE I: USA standards for sitting posture angles

Sitting Posture	USA standards
Forward head tilt	Maximum of 15°
Upper arms and forearms position	Elbow angles between 70° and 135°
Hands and wrists position	Maximum wrist extension (bent up) of 10° or flexion (bent down) of 30°
Vertical viewing angle	15–25° below the horizontal eye level
Shoulder angle	90°
Forward position	Adjustable max. of 330 mm from seat reference point
Backrest angle for swayback post.	Adjustable between 90° & 120° from horizontal
Upright Sitting angle	90°
Armrests height	Adjustable between 170 & 270 mm above seat

for comprehensive postural analysis: the head, both right and left shoulders, sternum, upper arms, forearms, hands, pelvic region, upper legs, lower legs, and feet. Each IMMU sensor in this system contains a 3D accelerometer, a 3D gyroscope, and a 3D magnetometer. The sensors are 34.5 (W), 57.8 (L), and 14.5 (H) mm in size, with a mass of 0.027 kg and powered by a battery. The Xsens software uses a Kalman filter to fuse the data of accelerometers, gyroscopes, and magnetometers to estimate the orientation of each sensor. The acquired 3D sensor data serves as the ground truth (GT) for keypoint locations and joint angles only for the Posture dataset, which is used to estimate the MSE between the predicted 3D keypoints and the joint angles relative to the GT. The acquired 3D sensor data is also annotated with one of the five sitting postures, which serves as the benchmark for evaluating the accuracy of posture recognition in the experiments carried out on the Posture Dataset.

During the data acquisition phase, seated at the workstation, the subject underwent five separate data collection sessions. Each session spanned 120 seconds. The Intel® RealSense™ system captured 1100 RGB and depth frames throughout these sessions. Concurrently, the Xsens system, functioning as a reference, recorded data at a sampling rate of 60Hz, resulting in a collection of 7200 samples per sensor. This dual-system approach facilitated a comprehensive and comparative analysis of the dataset. The dataset primarily comprises sitting postures, mainly consisting of (1) *Normal posture* characterized by sitting upright, with the back supported by the chair and hands on either the lap or the desk. (2) *The crouching posture* sits upright, with the neck bent downwards and the hands resting on the lap or desk. (3) *The hunchback* position involves sitting with a rounded back, a forward head close to the table, and hands either on the lap or the table. (4) *The slouching posture* is characterized by sitting with rounded shoulders and a forward head tilt. (5) *Swayback posture* refers to sitting with the hips in front of the body's midline. Table I shows various sitting postures and the ergonomics standards and guidelines for computer workstation design, as outlined [30].

2) Taekwondo Dataset: The dataset was developed specifically with data on movements performed by Taekwondo athletes. To achieve this task, we collected videos of students at Darimar Martial Arts, Columbus, Ohio. The acquired dataset comprises various Taekwondo patterns, each symbolizing a distinct movement executed by an athlete for a specific belt. The patterns include the following belt colors: white, yellow, orange, green, and black. Understanding patterns is a crucial component of Taekwondo training, as explained in the Taekwondo America student manual [29]. Patterns aid in the development of proper concentration and technique. The principal objective of this dataset is to gather data on the movements in any given belt pattern executed by Taekwondo athletes and to utilize this information for instructional and training purposes for students. We have a collection of 35 videos in total, which feature either a single student or multiple students performing the movements in sequence for each belt pattern. The videos were captured primarily using the camera on the phone. Fig. 5 (a) shows the walking stance low block, and (b) shows the walking stance reverse punch of a student in a dark green belt pattern.

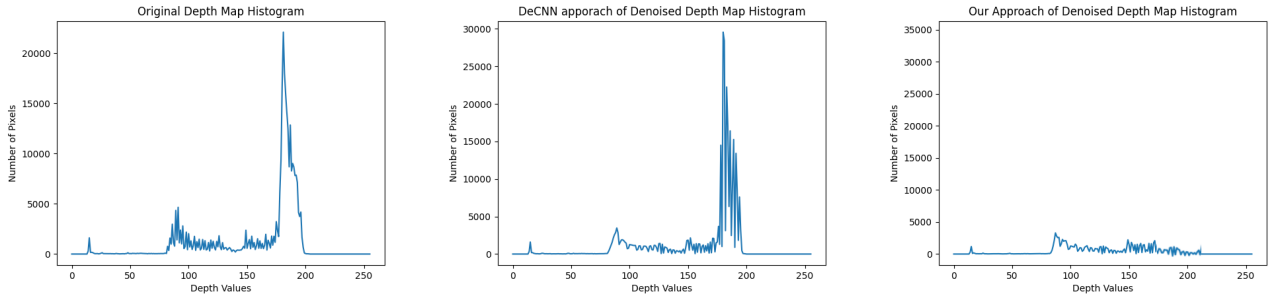
(a) (b)

Fig. 5: Green belt movement patterns (a) walking stance low block, (b) walking stance reveres punch

Table. II shows the accurate order of 20 movement patterns in the green belt, along with the validated angles and direction of the arm movements, as confirmed by the master of Darimar Martial Arts. A 10 to 15 degree deviation in body positioning is commonly considered as acceptable for posture detection and aligns with ergonomic studies, sports analysis, and motion tracking [31], as well as Taekwondo masters' advice for maintaining posture to optimize performance and prevent injuries. The vision algorithms themselves can usually achieve much better accuracies than 10-15 degrees; for instance, markerless motion capture systems employing vision algorithms attain angular accuracy of 1 to 3 degrees, ensuring minimal deviation between estimated and actual joint angles.

(a) (b) (c)

Fig. 6: Depth Transformation Results (a) Original depth image (b) DECNN, (c) Our depth transformation method



(a) (b) (c)

Fig. 7: Histogram Distribution of Depth after Depth Transformation (a) Original depth image (b) DECNN, (c) Our depth transformation method

TABLE II: Movements in Green Belt Pattern and the correct angles and direction of movements

Movement	Angle/Direction	Movement/Kick
1	90d Left	L Walking Stance
2	Forward	R Front Kick ->R Front Stance
3	180d Right	R Walking Stance
4	Forward	L Front Kick ->L Front Stance
5	90d Left	L Walking Stance
6	Forward	R Walking Stance
7	90d Left	R Back Stance
8	Hold Position	Shift Foot Into L Front Stance
9	180d Right	L Back Stance
10	Hold Position	Shift Foot Into R Front Stance
11	90d Left	L Walking Stance
13	270d Left	L Walking Stance
14	Forward	R Front Kick ->R Front Stance
15	180d Right	R Walking Stance
16	Forward	L Front Kick ->L Front Stance
17	90d Left	L Walking Stance
18	Forward	R Walking Stance
19	Forward	L Front Kick ->L Walking Stance
20	Forward	R Front Kick ->R Walking Stance

B. Depth Transformation Results

As mentioned in Section. IV-B, the depth maps obtained from RGB-D cameras are processed per frame to minimize the effect of noise. We have compared our methodology with a cutting-edge methodology in [26] where the authors have proposed a lightweight Convolutional Neural Network (CNN) specifically designed to eliminate noise and improve the quality of the depth map. The network comprises three layers that perform high-dimensional projection, missing data completion, and image reconstruction. It takes grayscale and depth images as inputs. The loss function utilizes an Euclidean-based distance metric to highlight the impact of edges, which quantifies the disparity between the network output and the corresponding ground truth. The drawback of this proposed DECNN framework is the amount of preprocessing required on the training data before feeding into the model. The preprocessing procedure has six steps: intensity equalization, bilateral filtering, edge extraction, watershed segmentation, segment average padding, and intensity quantization. After preprocessing, the unnecessary detail is weakened, and edges are enhanced.

Our depth transformation preprocessing includes two steps: (1) An offline alignment utilizing a bi-linear interpolation technique that relies on the scale factor between two images. It is necessary to perform this step because the field of view of the RGB and depth cameras differs in the Intel® RealSense™ D435i camera. The d435i depth camera has a field of view of 87° × 58°, while the RGB camera has a field of view of 69° × 42°, and (2) Applying a Fast Non-local means denoising. Fig. 6 (a), (b), and (c) show the raw depth image as captured from the depth camera and the results of enhancing depth maps using DECNN and our proposed method, respectively. Our depth transformation, explicitly targeting the person without significant preprocessing, has noticeably improved the spatial artifacts compared to the DECNN approach. Fig. 7 (a), (b), and (c) display the histogram of depth values for the raw depth image, the transformed depth images using DECNN, and our proposed method, respectively. The x-axis represents the depth values, while the y-axis represents the number of pixels with a specific depth value on the x-axis. The observed lack of sharp peaks in Fig. 7(c) provides a good indication that our results do not suffer from significant depth “holes”. Further, our proposed

approach to enhancing the depth map is corroborated by the much fewer black dots in Fig. 6(c).

C. 3D Keypoints and Joint Angle Estimation Results

Our proposed 3D posture recognition framework uses RGB-D input images. Using the depth information, a set of 2D joint locations and segment angles on the body are transformed into 3D. Fig. 8 (c) demonstrates that our proposed 3D posture recognition is highly accurate, as evidenced by the minimal deviations between the 3D key points obtained through our framework and the ground truth (GT) represented by the grey points and the line.

D. 3D Posture Recognition Results

We evaluate the performance of our proposed posture recognition using two metrics: (1) Accuracy in classifying the sitting postures with our proposed posture recognition framework, (2) Mean Squared Error (MSE) relative to ground truth (GT) for identifying 3D keypoints and joint angles, and (3) Accuracy in determining the correct sequence and posture of each pose during pose transitions using the pose transition framework. The GT is given by the Xsens system shown in Fig. 4 as discussed in section VI-A1.

We evaluate the accuracy and MSE of our posture recognition framework on a widely used public dataset called Human3.6M [32], which includes 3D information on joint angles, key points, and depth maps. The dataset is structured into 15 training motions, primarily focusing on sitting poses, particularly emphasizing categories such as sitting on a chair and engaging in various activities while seated, such as talking on the phone. We noticed that the poses captured for these activities also include the five categories of sitting postures we examined in this study. These postures can be utilized to assess the accuracy of our posture recognition framework.

1) Comparison with Machine Learning (ML) baseline models: We compare the performance of our proposed 3D Posture Detection framework with existing Machine Learning models as shown in Fig. 9(a), where the x-axis indicates the ML models used for comparison and the y-axis shows accuracy. We include different classifiers for comparisons like Decision Tree (DT), Support Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF), and Bagging Classifier (BC). The input to these ML models is the estimated 3D key points and the joint angles. This data is sampled every five seconds and is labeled into one of the five categories of sitting posture as explained in section VI-A1. We observe that the ensemble learning-based models are showing better results because they aggregate results of individual weak classifiers based on different strategies. Still, our satisfiability-based method outperforms all the ML baseline models.

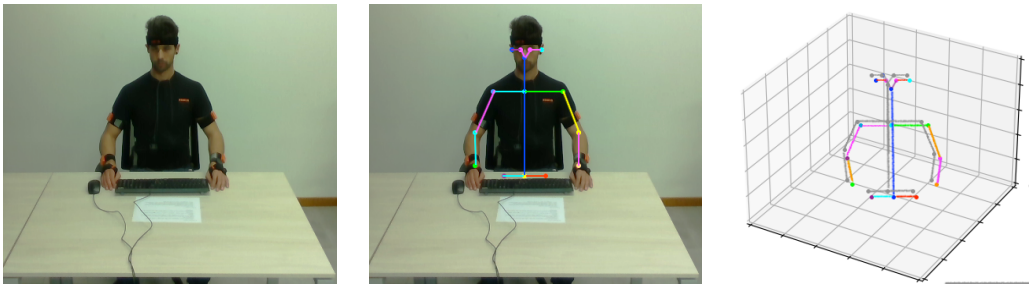
2) Comparison with time-series baseline models: To show the effectiveness of our proposed 3D posture detection framework on both the datasets, we compare them with several baselines that work with multivariate time series classification using sktime library [33] as shown in Fig. 9(b). We train the following time-series classifier models: Supervised Time Series Forest Classifier (STSF), Time Series Forest Classifier (TSFC), Time Series Support Vector Classifier (TSSVC), Random Interval Spectral Ensemble (RISE), Ensemble of Bag of Symbolic Fourier Approximation Symbols (BOSS). As shown in Fig 9 (b), our proposed posture detection framework achieves better accuracy than all of these time-series models.

3) Comparison with state-of-the-art 3D pose recognition models: For completeness, we present a qualitative analysis that compares our results against other 3D human pose estimation methods proposed in [19], [20], [34], and the performance results are shown in Fig 9 (c). In [34], authors have proposed a method to estimate 3D human pose from RGB-D images. The model comprises three modules. Initially, a 2D pose estimator generates heatmaps from the RGB image, serving as an initial prediction crucial for subsequent 3D estimation stages. Following this, the 2D fusion module integrates these heatmaps with the depth image to produce a point cloud, where each point is associated with a color feature vector. These downsampled points are inputted into a 3D learning module to generate point-wise features. Finally, a dense prediction module generates the 3D pose by point-wise voting. Ref [20] presents an anchor-based regression network for 3D hand and body estimation from a single depth image, which consists of 3 branches driven by a 2D CNN backbone network without deconvolutional layers. Specifically, the three branches are responsible for predicting in-plane offsets between anchor points and joints, estimating the depth value of the joints, and providing informative anchor point proposals. V2V-PoseNet [19] is a 3D hand and human pose estimation using a single depth map. This approach converts the 2D depth map into a 3D voxel representation and further processes it using a 3D CNN model, which predicts the per-voxel likelihood for each key point. The results in Fig. 9(a) and (c) show that 3D posture recognition approaches work better than traditional ML approaches because they can handle sequential data and take into account how data changes over time. *Yet, our proposed satisfiability-based approach recognizes posture with the highest accuracy compared to ML, time series, and state-of-the-art 3D posture recognition models.*

4) MSE and MPJME of Proposed 3D Posture Recognition Framework: To assess the accuracy of our 3D posture recognition approach to the existing ground truth in both the posture dataset and the Human3.6M data, we present a qualitative analysis that compares our results against other 3D human pose estimation methods in terms of Mean Squared Error (MSE), which is the relative mean square error between the ground truth 3D coordinate of the joints with respect to the estimated 3D coordinates of the joints defined as:

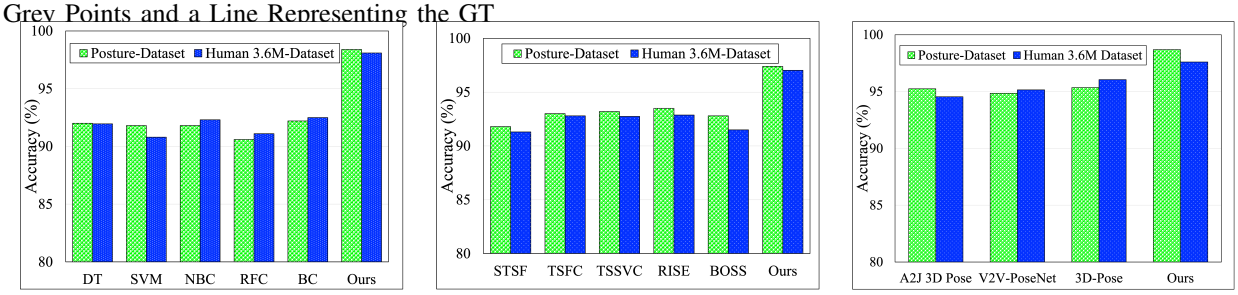
$$MSE = \frac{1}{N}\sum_{n=1}^{N}\sum_{i,j,k}\|H_n^*(i,j,k) - H_n(i,j,k)\|^2 \qquad (5)$$

Where $H_n^*(i,j,k)$ and $H_n(i,j,k)$ are the ground-truth and estimated keypoint locations for nth key point, respectively, and N denotes the number of key points. Further, we also use Mean Per Joint Position Error (MPJPE), a common metric used to evaluate the performance of human pose estimation algorithms. It measures the average distance between the predicted joints of a human skeleton and the ground truth joints. The results of both MSE and MPJME are presented



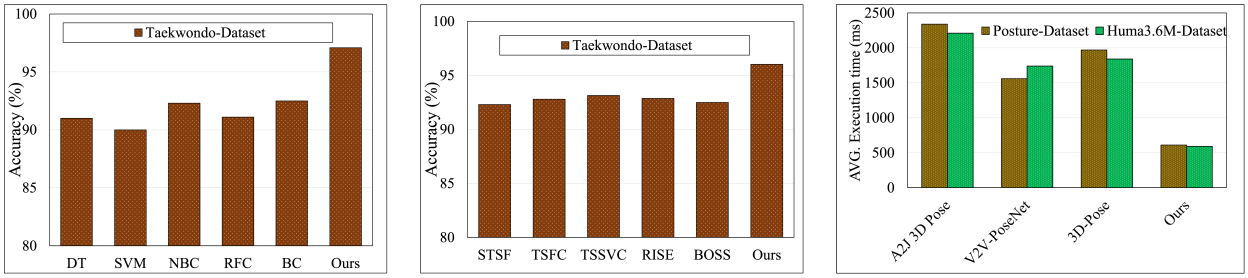
(a) (b) (c)

Fig. 8: 3D Posture Recognition Results Compared to GT (a) Original RGB-Image, (b) 2D Keypoints, (c) 3D Estimation Results with Grey Points and a Line Representing the GT



(a) (b) (c)

Fig. 9: Comparison of Accuracy of proposed 3D posture detection framework on both datasets to (a) ML classifier models, (b) Time-series classifier models, and (c) State-of-the-Art 3D posture recognition models



(a) (b) (c)

Fig. 10: Comparison of Accuracy of pose transition framework on Taekwondo dataset to (a) ML classifier models, (b) Time-series classifier models, (c) Average Running Time Comparison

in Table III. We can see that our approach's performance is better than other methods, validating the effectiveness of our 3D posture recognition framework.

E. Pose Transition Results

We assess the efficiency of our proposed pose transition model by measuring its accuracy in accurately determining the sequence and the correctness of each posture during pose transitions. Each frame in the Taekwondo dataset is assigned a label indicating the transition between poses, which corresponds to a specific belt pattern. Additionally, the label indicates whether the posture in the frame is proper or improper. Since we have already demonstrated accurate pose identification in the previous application, our goal here is to determine the accuracy of the pose transition framework.

We compare our pose transition framework to the traditional ML and the time-series models considered in Sections VI-D1 and VI-D2. For the pose transition framework, the input to the state-of-the-art ML/DL or time-series models are frame-wise labeled postures and the movements in specific belt patterns(as shown in Table II to identify the correct sequence of posture transitions. The SMT assertion to classify each pose during pose transition in this green belt movement pattern is shown in Definition 2. Additionally, a decision function is used to determine if the sequence of posture transitions corresponds to a specific belt pattern. Fig. 10 illustrates the accuracy of our pose transition framework in comparison to traditional machine learning (ML) and time series (TS) models. The x-axis represents the different models being considered, while the y-axis represents the corresponding accuracy values. Our proposed approach demonstrates better accuracy compared to all the machine learning (ML) and time series (TS) models that were considered.

Fig. 10(c) shows the inference time of our framework for both datasets against the state-of-the-art 3D posture recognition methods considered in section VI-D3. Our reported time includes OpenPose-based key point detection, estimation of joint angles, and checking for satisfiability in classifying the postures. For deep learning-based models, the inference time is used for comparison. It is seen that our inference

TABLE III: Comparison of MSE of 3D human pose estimation

Methods	Datasets	MSE	MPJME
A2J 3D Pose	Posture	32.5	42.3
	Human 3.6M	30.7	40.8
V2V-PoseNet	Posture	36.3	45.3
	Human 3.6M	31.4	43.1
3D-Pose	Posture	34.35	45.2
	Human 3.6M	34.1	41.4
Ours	Posture	21.2	36.3
	Human 3.6M	22.5	34.5

time is the lowest. The time DL models require to classify postures is proportional to the number of layers and the model size. However, our proposed framework uses DL (OpenPose model) solely for body joint key points detection, thereby maintaining a decent inference time to recognize the proper or improper posture. The average time taken by our framework to acknowledge the proper or improper postures for posture and taekwondo datasets is 610 and 590 ms, respectively. It is essential to mention that these experiments were conducted on a machine simulating an edge device, not an edge controller (EC) that typically has much higher processing capabilities.

F. Running Time Comparison

Furthermore, analyzing every frame is unnecessary for both of these use cases. Instead, we can use an approach similar to the one in [35] to filter out frames based on the change and send only a subset to the EC. Thus, with only $\sim$10 frames/sec, real-time posture recognition/ pose transitions may be possible. However, activities involving rapid movements, such as gymnastics or dance, would require much higher resources for real-time posture monitoring.

VII. Conclusions

In this paper, we studied the problem of detecting correct posture and pose transitions, which are crucial in numerous human activities, from sitting in front of a computer screen to sports and performing arts. We build a spatio-temporal reasoning infrastructure on top of traditional computer vision algorithms to recognize and analyze the poses and pose transitions. For the pose, we use a 2D human stick model, openpose, and enhance it further to determine the depth and relevant 3D angles, which can be compared against the existing standards. We specifically consider the examples of sitting and Taekwondo and show that our method outperforms all machine learning, deep learning, and time series-based methods in both accuracy and execution time despite not needing any specialized training data. Our method does require setting up the assertions. This task can be automated to a large extent, and will be examined in the future.

References

[1] B. Channel, "The dangers of sitting," https://www.betterhealth.vic.gov.au/health/healthyliving/the-dangers-of-sitting, April 2022.

[2] P.-C. Lin *et al.*, "Automatic real-time occupational posture evaluation and select corresponding ergonomic assessments," *Scientific Reports*, vol. 12, no. 1, p. 2139, 2022.

[3] D. Kee, "Systematic comparison of owas, rula, and reba based on a literature review," *IJERPH*, vol. 19, no. 1, p. 595, 2022.

[4] F. Ghasemi *et al.*, "A new scoring system for the rapid entire body assessment (reba) based on fuzzy sets and bayesian networks," *Int. J. Ind. Ergon.*, vol. 80, p. 103058, 2020.

[5] J. Meyer *et al.*, "Design and modeling of a textile pressure sensor for sitting posture classification," *IEEE Sensors Journal*, vol. 10, no. 8, pp. 1391–1398, 2010.

[6] K. Ishac *et al.*, "Lifechair: A conductive fabric sensor-based smart cushion for actively shaping sitting posture," *Sensors*, vol. 18, no. 7, p. 2261, 2018.

[7] T. D. Nguyen *et al.*, "A survey of top-down approaches for human pose estimation," *arXiv preprint arXiv:2202.02656*, 2022.

[8] G. H. Martınez, "Openpose: Whole-body pose estimation," Ph.D. dissertation, Carnegie Mellon University Pittsburgh, PA, USA, 2019.

[9] K. Chen, "Sitting posture recognition based on openpose," in *IOP Conference : MSE*, vol. 677, no. 3. IOP Publishing, 2019, p. 032057.

[10] S. Mroz *et al.*, "Comparing the quality of human pose estimation with blazepose or openpose," in *BioSMART*. IEEE, 2021, pp. 1–4.

[11] C. Zheng *et al.*, "Deep learning-based human pose estimation: A survey," *arXiv preprint arXiv:2012.13392*, 2020.

[12] H. Xu *et al.*, "Ghum & ghuml: Generative 3d human shape and articulated pose models," in *IEEE/CVPR*, 2020, pp. 6184–6193.

[13] Y. Ming *et al.*, "Deep learning for monocular depth estimation: A review," *Neurocomputing*, vol. 438, pp. 14–33, 2021.

[14] G. Moon *et al.*, "Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image," in *Proceedings of the IEEE/CVF*, 2019, pp. 10 133–10 142.

[15] J. Y. Chang *et al.*, "Poselifter: Absolute 3d human pose lifting network from a single noisy 2d human pose," *arXiv preprint arXiv:1910.12029*, 2019.

[16] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *CVPR 2011*. Ieee, 2011, pp. 1297–1304.

[17] R. Girshick *et al.*, "Efficient regression of general-activity human poses from depth images," in *2011 International Conference on CV*. IEEE, 2011, pp. 415–422.

[18] J. Gall *et al.*, "Hough forests for object detection, tracking, and action recognition," *IEEE TPAMI*, vol. 33, no. 11, pp. 2188–2202, 2011.

[19] G. Moon *et al.*, "V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map," 2018.

[20] F. Xiong *et al.*, "A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image," in *Proceedings of the IEEE/CVF*, 2019, pp. 793–802.

[21] L. Needham *et al.*, "The accuracy of several pose estimation methods for 3d joint centre localisation," *Scientific reports*, vol. 11, no. 1, p. 20673, 2021.

[22] Z. Cao *et al.*, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields. arxiv 2018," *arXiv preprint arXiv:1812.08008*, 1812.

[23] M. M. Ibrahim *et al.*, "Depth map artefacts reduction: A review," *IET Image Processing*, vol. 14, no. 12, pp. 2630–2644, 2020.

[24] P.-L. Liu *et al.*, "Simple method integrating openpose and rgb-d camera for identifying 3d body landmark locations in various postures," *Int. J. Ind. Ergon.*, vol. 91, p. 103354, 2022.

[25] K. Lee *et al.*, "Efficient depth enhancement using a combination of color and depth information," *Sensors*, vol. 17, no. 7, p. 1544, 2017.

[26] X. Zhang *et al.*, "Fast depth image denoising and enhancement using a deep convolutional network," in *ICASSP*. IEEE, 2016, pp. 2499–2503.

[27] L. De Moura *et al.*, "Satisfiability modulo theories: Introduction and applications," *Commun. ACM*, vol. 54, no. 9, pp. 69–77, 2011.

[28] L. de Moura *et al.*, "Z3: An efficient smt solver," in *TACAS*, C. R. Ramakrishnan *et al.*, Eds., 2008, pp. 337–340.

[29] "Taekwondo student manual," https://taekwondoamerica.org/wp-content/uploads/2017/09/Student-Manual-2012.pdf, accessed: 20 December 2023.

[30] E. Woo *et al.*, "Ergonomics standards and guidelines for computer workstation design and the impact on users' health–a review," *Ergonomics*, vol. 59, no. 3, pp. 464–475, 2016.

[31] W. G. Janssen *et al.*, "Determinants of the sit-to-stand movement: a review," *Physical therapy*, vol. 82, no. 9, pp. 866–879, 2002.

[32] C. Ionescu *et al.*, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE TPAMI*, vol. 36, no. 7, pp. 1325–1339, jul 2014.

[33] M. Löning *et al.*, "sktime: A unified interface for machine learning with time series," *arXiv preprint arXiv:1909.07872*, 2019.

[34] J. Ying *et al.*, "Rgb-d fusion for point-cloud-based 3d human pose estimation," in *2021 IEEE ICIP*. IEEE, 2021, pp. 3108–3112.

[35] P. Pradeep *et al.*, "Resource efficient edge computing infrastructure for video surveillance," *IEEE Trans on Sustainable Computing*, March 2021.