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

Pavana Pradeep Kumar*, Krishna Kant*, Francesco Di Rienzo†, Carlo Vallati†
*Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
†Department of Information Engineering, University of Pisa, Italy
Email: {pavana.pradeep, kkant}@temple.edu, francesco.dirienzo@phd.unipi.it, carlo.vallati@unipi.it

***Abstract*—Correct pose/posture is crucial in most human activities, and increasingly in using computer screens of many form factors. In performing arts, both the correct pose and correct pose transitions are important for acceptable performance and safety. We show in this paper that the pose-relevant angles can be estimated in real-time and accurately from RGB-D camera views, using a spatio-temporal reasoning infrastructure on top of basic computer vision algorithms. The latter is used to determine poses of the 2D human stick models, which are then enhanced further using depth to determine relevant angles and compare them against the standards. We apply the technique for the case of a knowledge worker sitting in front of a computer screen and also to tae-kwando, where the pose transitions are also crucial. In either case, our method does not require any additional training but outperforms all others, including those using machine learning, deep learning, and time series based prediction. Furthermore, the superior performance is seen not only in the estimation accuracy but also the estimation speed.**

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

I. Introduction

Correct pose/posture is crucial in numerous different contexts and may be important from many different perspectives such as aesthetics (e.g., dancing), successful performance of an activity (e.g., gymnastics), avoiding serious 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 some cases, it is not just the pose, but pose transition that is also crucial (e.g., dancing, martial arts, gymnastics, etc.)

The goal of this paper is to demonstrate that we can accurately determine the pose-relevant angles and dimensions accurately from RGB-D camera views, and thereby enable comparison of the pose and pose transition against the given standards or requirements. To accomplish this, we build a spatio-temporal reasoning infrastructure on top of basic computer vision algorithms that detect objects (humans) and construct 2D stick models of their poses. We consider two distinct applications in this paper. Our first application is a knowledge worker sitting in front of a computer monitor. In this case, we obtained highly accurate and detailed ground-truth measurements about the posture using a set of senors. By comparing our results against these, we demonstrate that our video analytics method provides highly estimates of various angles. Our second application is Tae-kwando where we annotated the frames manually to allow for accuracy estimation. In this application, both the pose and the sequence of pose transitions is crucial. In either case, our method does not require any training on the labelled postures, since it use pretrained OpenPose as an underlying model, whose output is further processed to accurately estimate the depth dimension and from there estimates of the angles.

This research was supported by NSF grant CNS-1527346.

The rest of the paper is organized as follows. Section II discusses the existing work-related posture standards and posture (pose) modeling in the computer vision context. Section III defines our methodology for modeling and tracking the posture. Section IV presents the experimental results, and then section V concludes the discussion.

II. Posture Standards and Modeling

A. Posture Standards

Human posture in the context of ergonomics has been studied extensively. Posture very much depends on factors such as chair height, contact with the backrest, screen height/distance, etc.; thus, these also become part of various ergonomics standards. such as those defined for USA, Australia, Europe, and Japan [2]. Several methods have been developed for assessment of body poses for various types 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 briefly in the following.

Rapid Entire Body Assessment (REBA) (https://ergo-plus.com/reba-assessment-tool-guide/) uses a systematic way to evaluate whole body postural MSD and ergonomic design risks associated with job tasks. REBA was intended to evaluate required body posture, forceful exertions, type of movement or action, repetition, and coupling by manually filling out a form. A score is assigned for each of the following body regions: wrists, forearms, elbows, shoulders, neck, trunk, back, legs and knees. After the data for each region is collected and scored, tables on the form are then used to compile the risk factor variables, generating a single score that represents the level of MSD risk.

Rapid Upper Limb Assessment (RULA) (https://ergo-plus.com/rula-assessment-tool-guide/) assesses biomechanical and postural load requirements of job tasks/demands on the neck, trunk and upper extremities. RULA also involves filling out a form manually to evaluate required body posture, force, and repetition. Based on the evaluations, scores are entered for each body region in section A for the arm and wrist, and

section B for the neck and trunk. It too generates a single score for MSD risk based on this data.

WISHA Lifting Calculator (https://ergo-plus.com/wisha-lifting-calculator-guide/) was developed by the Washington State Department of Labor and Industries and is based on NIOSH research on the primary causes of back injuries. This lifting calculator can be used to perform ergonomic risk assessments on a wide variety of manual lifting and lowering tasks, and can be also used as a screening tool to identify lifting tasks which should be analyzed further using the more comprehensive NIOSH Lifting Equation.

Changes in posture provide valuable information about stress and attention level. Several gadgets have also been developed for posture monitoring such as textile pressure sensor [5], a sensored smart cushion for the chair back [6], an "IoT cushion" and AI based posture training [7].

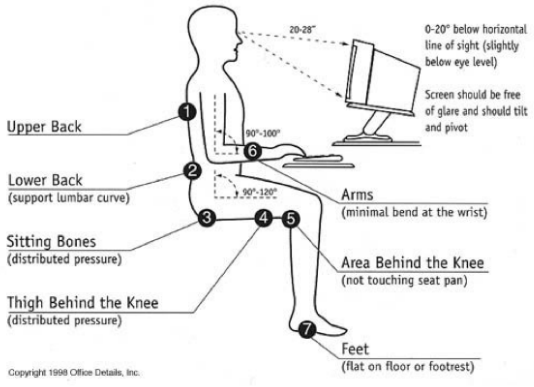


Fig. 1. Sitting Posture Recommendations

Recently, there has been attempt to apply these standard to video determined postures. There are 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/, however, it is unclear how the tools work. However, video based determination becomes a bit difficult because of lack of completeness in the standards. In particular, while some angles are specified, several others are not, as shown in Fig II-A. In these cases, we make some assumptions. In particular, qualitative specifications like "back should be straight" do not provide tolerance; therefore, we assume a tolerance value value of $\pm 10^{\circ}$.

B. Video Based Posture Modeling

Accurately recognizing posture from videos requires 3D structure of the body, which is difficult to obtain from regular RGB cameras in spite of a significant amount of work to estimate depth from plain RGB frames or "monocular" images [8], [9]. These methods must necessarily depend on rather complex deep learning and thus have poor accuracy and are too slow for real-time use. A direct estimation of depth requires either a scanning LIDAR or depth (or RGB-D) cameras. RGB-D cameras are becoming quite popular along with a continuing decrease in prices. We expect that RGB-D cameras would become commonplace and thus support depth estimation in work environments. Nevertheless, much of work in the literature concerns RGB cameras. Our approach is to enhance known 2D methods with our unique depth processing to extract 3D information.

Recognizing humans and other objects and tracking them in successive RGB camera video frames is very well developed art. For real-time, use, the single pass algorithms such as SSD (single-shot multibox) versions [10] and YOLO (you only look once) versions and including the recent YOLOv7 [11] are most appropriate. Object tracking is rather straightforward so long as the object remains in view, but persistent tracking can be challenging and largely studied using rather heavy-duty deep learning [12]. Although our logic based method along with object attributes can easily do persistent tracking, we do not focus on this aspect.

Human pose estimation based on images is a very well developed area, with several pose models and algorithms. 2D pose models assume a "stick model" of human and recognize individual segments of the model. The algorithms can be classified as top-down (detect each object and then its parts), bottom-up (detect all parts and then assemble into objects), or combined [13]. Bottom-up methods are faster than top-down when the scene has many people, but may assign parts to the wrong people. YOLOv7 mentioned above is a top-down method and seems to work well. The survey paper [14] provides a comprehensive overview of popular top-down approaches for human pose estimation. A highly popular botttom-up method is OpenPose [15]. It define a number of keypoints in the body to create a skeleton that represents various poses. Sitting posture recognition based on OpenPose is discussed in [16]. We shall use OpenPose in this research. Other bottom-up methods include Higher-HRNet [17], Posenet, and Movenet [18]. Note that these algorithms recognize only human objects and thus need to be used with others to parse the entire image.

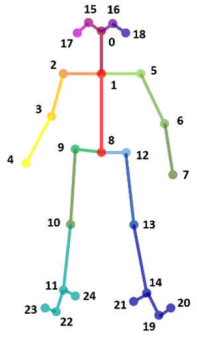


Fig. 2. Body_25 Model of OpenPose

As such OpenPose only concerns the 2D image (i.e., no depth). Furthermore, the body segments are simply sticks without any width information, thus the width also needs to be determined by further analysis of the 2D image.[1] The depth information is contained in the in the "3D point cloud" generated by both RGB-D camera and scanning LIDAR. Depth maps generally have a lot of defects and noises due to varying levels of ambient illumination and light diffraction/reflection from various surfaces, edges, and points. Numerous methods have been investigated to curate the depth-map and integrate it with RGB image, but all previous methods concern smoothing the depth map without of with the help of the RGB image and involve some form of deep learning [19]–[21]. Reference [22] evaluates

[1]OpenPose does use a single width hyper-parameter for all limbs, but that is not useful.

the relative accuracy of various 3D pose estimation methods. Reference [23] provides a survey of various methods.

It is worth noting that although true 3D (human) body/pose models (as opposed to stick models) do exist [13], [24], they are not usable for real-time recognition and will be deeply affected by clothing. The 2D models themselves vary; for example, called COCO (or body_17) does not have key points for hands/feet, but the body_25 human model shown in Fig. 2 does. Another model called Blazepose [25] is even more detailed than body_25. In this paper, we use body_25; it is possible to similarly use others for applications where precise pose of hands/fingers and feet/toes is important.

III. Proposed System for Active Posture Monitoring

The proposed system for active posture monitoring gets 3D body joint locations using OpenPose and an RGB-D camera. This tests the usefulness of the suggested method for measuring landmarks from a range of common postures. The anatomical 3D body landmark positions are identified by 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 IV-A were used as references to quantify the error levels. This study uses an RGB-D camera with active stereoscopic technology (i.e., Intel® RealSense™ D435i).

A. OpenPose based Keypoints Extraction

Fig. 3 illustrates the stages involved in our 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.

To understand how OpenPose determines the angles of various limbs, we briefly describe its functioning. OpenPose uses the first 10 layers of the popular deep convolutional network VGG-19 [26] to extract features before proceeding with pose detection. For the latter, the OpenPose network uses two branches [27], each with $K = 6$ *stages*. One branch estimates the *Confidence Map* (CM), which gives the keypoint locations of each object. The second branch estimates the *Part Affinity Field* (PAF), which gives orientations of all the limbs.

During inferencing, OpenPose has the keypoint CMs of all the agents and limb orientations as PAFs which provides the XY angles. Because of its bottom-up nature, the association between limbs and people present in the scene. This 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. This also implies that in crowded scenes, the limb assignment and hence the angles could be incorrect; however, this should not be an issue in most posture applications. As for 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 camera has a somewhat larger focal length than the RGB camera, and thus the images do not overlap. Furthermore, the depth of the image quality is not good enough to run OpenPose on it and accurately locate the corresponding key points. For this work, we have used an ad hoc correction method since the discrepancy does not change across different frames. After alignment, we take the raw depth measure at the point on a limb in the RGB image and then curate the depth to remove noise. We also apply the inverse projection transformation to transform the image coordinates into the object coordinates.

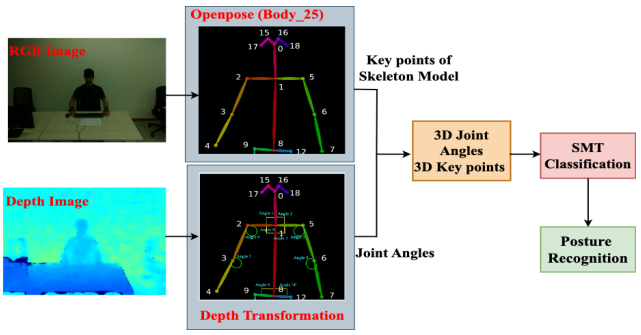


Fig. 3. Posture Recognition from RGB-D cameras

B. Depth Transformation

1) Depth Map Correction: Depth maps captured from RGB-D cameras consist of Gaussian noise and hole pixels, which can be categorized as spatial artefacts, along with some temporal fluctuations, which can be categorized as temporal artefacts. Hence, the input depth map must first be processed to reduce such artefacts. There are many approaches for noise reduction in depth maps, 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]).

Instead of following a traditional approach to reducing noise in depth maps, we apply a more sophisticated approach to improve the quality of depth maps. Our approach for reducing the noise and filling the region of hole pixels involves estimating depth limb-wise. 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 estimate the depth and, hence, the angle by assuming that the arm is straight and roughly 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) as estimated above.

Let θ be the depth gradient to be estimated, and $d(l,w)$ the measured depth at a point **(pixel)** at distance $l \in [0..L]$ from keypoint 2 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 depth at every point in the arm to a single point by using the following transformation:

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

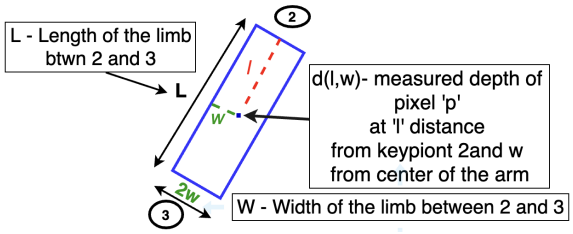


Fig. 4. Estimation and Curation of Depth

That is, $d'(l,w) = C$, where C is a constant independent of l and w. Now, 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 center of the cluster as the true depth. Practically, this requires finding the cluster's center using all the data, removing points falling outside some 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 start with the depth at keypoint 2, which is already known. This successive procedure ensures joint consistency, although more sophisticated procedures can also be used, which will be seen in future works.

C. Extracting 3D Joint Angles and Keypoints

OpenPose is a highly reliable tool for extracting skeletal structures not trained on pre-defined body poses. It isolates each joint from the overall body pose. In this work, we utilize OpenPose to extract skeletal joint coordinates, which returns the 2D coordinates (x_i, y_i, c_i) for i = 0,1,..24 from an RGB image using CMs and PAFs; x_i and y_i are the abscissas and ordinates respectively of each 24 BODY_25 body parts, while c_i represent their confidence measure. Hence running OpenPose on the RGB images collected from both the datasets described in the Section.IV-A yields a vector of 2D skeleton coordinates of 25 human body joints. However, we exclusively focus on 19 distinct key points for this study, thereby omitting the remaining key points (17–24). As described in the previous section, the calculated limb-wise transformed depth values are concatenated with the 2D skeleton coordinates, resulting in a 3D vector (x_i, y_i, d'_i).

Further, any two joints form one skeleton segment. And we have a total of 18 skeletal segments that are defined as S_i = $\{S_1, S_2.., S_{18}\}$. Each skeletal segment S_i consists of two joint points; for example, S_1 comprises joints $\{j_0, j_1\}$ where j_0 is keypoint joint 0 and j_1 is keypoint joint 1, as shown in Fig. 2. The spatial coordinates of two joints in any skeletal segment are expressed as a 3D vector expressed as $j_a = \{x_a, y_a, z_a\}$ a = 1,2...18 and $j_b = \{x_b, y_b, z_b\}$ b = 1,2...18, b$\neq$a. Then the direction vector of the linear equation of skeletal segment S_i is denoted as follows:

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

Thus the angle between the two skeletal segments S_a and S_b is defined as:

$$Angle = arcos\left(\frac{v_{xa} * v_{xb} + v_{ya} * v_{yb} + v_{za} * v_{zb}}{\sqrt{v_{xa}^2 * v_{ya}^2 + v_{za}^2} + v_{xb}^2 * v_{yb}^2 + v_{zb}^2}\right) \quad (4)$$

D. Satisfiability Modulo Theory (SMT) based Posture Recognition

Satisfiability involves determining whether a formula expressing a constraint has a solution. The popular constraint satisfaction problem has mainly has two variants, 1) *SAT*, which determines whether a formula composed of Boolean variables connected by logical conjunctions can be converted to true by selecting true or false values for its variables, and 2) *SMT* involves testing the satisfiability of first-order formulas over linear integer or real arithmetic, or other theories. A first-order formula combines logical connectives, variables, quantifiers, function symbols, and predicate symbols. A satisfiable solution, also called a model, interprets the variable, function, and predicate symbols that satisfy the formula.

We formulate our posture recognition mechanism as a Boolean satisfiability problem using the famous SMT (Satisfiability Modulo Theory) [28] based tools and linear arithmetic theory. Since we have five different poses, as explained in Section IV-A, pose recognition is a multi-class classification problem. Each pose, depicted in Fig 3, comprises certain angles. Various combinations of angles can be used to determine if a pose is incorrect and falls into one of the categories of wrong poses. Each combination of angles for a particular pose can be expressed as first-order logic formulas/assertions along with using relational symbol operators for equality and inequalities ($=, >, <, \leq, \geq$) are used to form atomic predicates. The SAT core returned from the SMT solver indicates the presence of a particular posture.

Each pose is categorized according to the value of joint segment angles within the unfavorable range. 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 angle 1 and angle 3, which indicate a slight

forward bending of the head, along with angle 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 assertion, which is given as input to Z3 [29], a popular SMT tool, demonstrates the classification of slouching:

(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

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

IV. 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.

(a) (b)

Fig. 5. (a) Experimental Setup, (b) Subject with Xsens motion capture sensors.

1) Posture Dataset: The data collection was meticulously designed to replicate an authentic office work environment, focusing on a worker engaged in computer-related tasks. The participant was outfitted with the Xsens motion capture system comprising 17 strategically placed sensors. These sensors were located at key anatomical landmarks for comprehensive postural analysis: the head, both right and left shoulders, sternum, upper arms, forearms, hands, pelvic region, upper legs, lower legs, and feet, as depicted in Fig. 5(b). 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. 5 (a).

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

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 maximum of 330 mm from seat reference point
Backrest angle (for swayback posture)	Adjustable between 90° and 120° from the horizontal
Upright Sitting angle	90°
Armrests height	Adjustable between 170 and 270 mm above seat

athletes. To achieve this task, we collect 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 comprise 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 [31]. 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. 6 (a) shows the walking stance low block, and (b) shows the walking stance reverse punch of a student in a dark green belt pattern.

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. When a student performs under the supervision of an expert, any deviation of 10 to 15 degrees in the angle of the positioning of body parts is considered.

B. Depth Transformation Results

As mentioned in Section. III-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 Convolutional Neural Network (CNN) for denoising depth maps called DECNN [21]. 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

(a) (b)

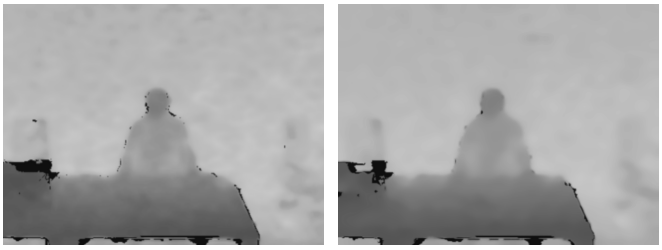
Fig. 6. Green belt movement patterns (a) walking stance low block, (b) walking stance reveres punch.

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
12	Forward	R 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

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 pre-processing procedure has six steps: intensity equalization, bilateral filtering, edge extraction, watershed segmentation, segment average padding, and intensity quantization. After pre-processing, the unnecessary detail is weakened, and edges are enhanced.

On the other hand, our depth transformation pre-processing includes two steps: (1) An offline alignment by 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. 7 (a) and (b) show



(a) (b)

Fig. 7. Depth Transformation Results (a) DECNN, (b) Our depth transformation method.

the results of enhancing depth maps using DECNN and our proposed method, respectively.

C. Posture Classification and Posture Transition Results

We evaluate the effectiveness of our proposed posture recognition model by measuring its accuracy in determining the correctness of a given posture and the accuracy in identifying the correct sequence of pose transitions. The Xsens system records 3D sensor data from 17 sensors placed on the subject's body, as depicted in Fig. 5(b). The sensor data is subsequently labeled with one of the five sitting postures and used as the reference for the experiments conducted on the Posture Dataset. Our posture recognition framework processes each input frame individually and calculates the 3D values of body joint locations and the angles connecting any two body joint key points using OpenPose. This 3D output is compared to the ground truth data to assess the accuracy of posture recognition. In addition, in the Taekwondo dataset, each video frame is labeled a pose transition corresponding to a specific belt pattern.

We have compared our posture recognition and pose transition framework against cutting-edge machine learning, time series, and deep learning models. The input for these models used in posture recognition consists of the 3D coordinates of body joint locations and the corresponding angles, categorized into one of the five sitting postures. However, the input to these state-of-the-art models is frame-wise labeled pose transitions for identifying the right sequence of posture transitions. Additionally, a decision function is used to determine if the sequence of posture transitions corresponds to a specific belt pattern.

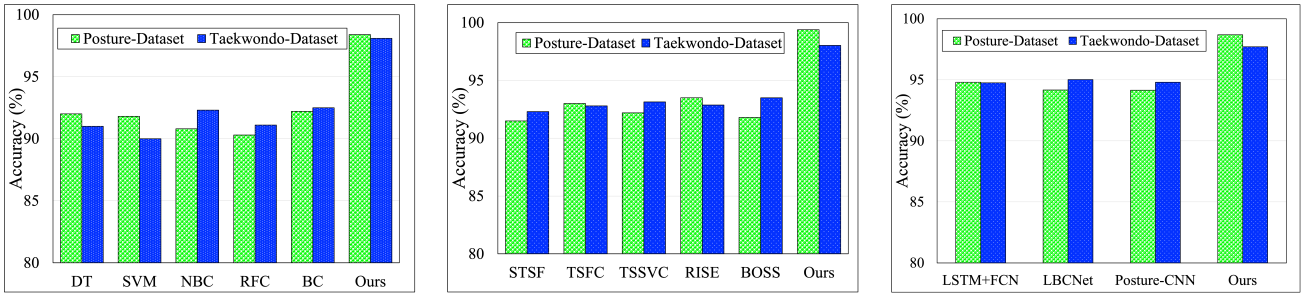
1) Comparison with Machine Learning (ML) baseline models: We compare the performance of our proposed Posture Detection framework with existing Machine Learning models as shown in Fig. 8(a) where the x-axis shows the ML models used for comparison and the y-axis shows accuracy. We include different classifiers for comparisons like Decision Tree (DT), Support Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF), and Bagging Classifier (BC). The collected sensor data are labeled into one of the five categories of posture recognition. We observe that the ensemble learning-based models are showing better results because they aggregate results of individual weak classifiers based on different strategies, and our satisfiability-based method outperforms the ML baseline models.

2) Comparison with time-series baseline models: To show the effectiveness of our proposed posture detection framework on both the datasets, we compare them with several baselines that work with multivariate time series classification using sktime library [32] as shown in Fig. 8(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 8 (b), our proposed posture detection framework achieves better accuracy than the above-mentioned time-series models.

3) Comparison with state-of-the-art Deep Learning (DL) baseline models: We compare our proposed model to three other existing posture recognition and prediction models proposed in [33]–[35], and the performance results are shown in Fig 8(c). Authors in [33] developed a system that combined two ultrasonic sensors and 16 pressure sensors. The collected signals were fused and processed by an Arduino board and then transmitted to a cloud platform, where a Convolutional Neural Network (CNN) and Lower-Balanced Check Network (LBCNet) were used for posture classification. The posture classification framework developed in [34] is a combined deep learning model consisting of a Fully Convolutional Network (FCN) and a Long Short-Term model (LSTM). The data was collected for nine different postures using three tri-axial accelerometer sensors placed on the backs of the subjects to monitor their posture during sitting and standing. In [35], authors have proposed a Posture-CNN deep learning method in the field of posture recognition. Posture-CNN can effectively reduce network parameters and improve network speed. Six distinct postures were monitored during data collection: walking, raising the left arm, the right arm, the arms, squatting, and lifting the legs. Human skeleton data is monitored using the Kinect V2.0 depth sensor's skeleton tracking function.

The results in Fig. 8(a) and (c) show that DL-based approaches work better than traditional ML approaches because they can handle sequential data and take into account how data changes over time. In contrast, our proposed model, our satisfiability-based approach, detects posture with the highest accuracy compared to ML, DL, and time-series models.

4) Running Time Comparison: Fig. 9 compares the inference time of our framework for both datasets against state-of-the-art DL-based posture-recognition methods considered in section IV-C3. For our framework, the reported time includes OpenPose based key point detection, estimation of joint angles, and checking for satisfiability to classify the postures. For DL models, it is inference time for classification. It is seen that our inference 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 keypoints detection, thereby maintaining a decent inference time to recognize the proper or improper posture. The average time taken by our framework to recognize the proper or improper postures for posture and taekwondo datasets is 610 and 590



(a) (b) (c)

Fig. 8. Comparison of Accuracy of proposed Posture detection framework to (a) ML classifier models, (b) Time-series classifier models, and (c) State-of-the-Art DL models.

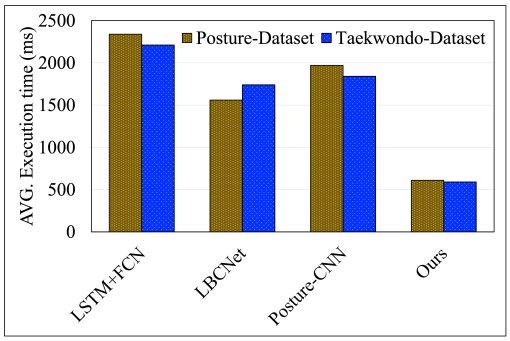


Fig. 9. Average Running Time Comparison

ms, respectively.

V. Conclusions

References

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

[2] P.-C. Lin, Y.-J. Chen, W.-S. Chen, and Y.-J. Lee, "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," *International Journal of Environmental Research and Public Health*, vol. 19, no. 1, p. 595, 2022.

[4] F. Ghasemi and N. Mahdavi, "A new scoring system for the rapid entire body assessment (reba) based on fuzzy sets and bayesian networks," *International Journal of Industrial Ergonomics*, vol. 80, p. 103058, 2020.

[5] J. Meyer, B. Arnrich, J. Schumm, and G. Troster, "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 and K. Suzuki, "Lifechair: A conductive fabric sensor-based smart cushion for actively shaping sitting posture," *Sensors*, vol. 18, no. 7, p. 2261, 2018.

[7] K. Bourahmoune, K. Ishac, and T. Amagasa, "Intelligent posture training: Machine-learning-powered human sitting posture recognition based on a pressure-sensing iot cushion," *Sensors*, vol. 22, no. 14, p. 5337, 2022.

[8] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 11, pp. 3174–3182, 2017.

[9] Y. Ming, X. Meng, C. Fan, and H. Yu, "Deep learning for monocular depth estimation: A review," *Neurocomputing*, vol. 438, pp. 14–33, 2021.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015. [Online]. Available: http://arxiv.org/abs/1512.02325

[11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[12] Y. Zhang, T. Wang, K. Liu, B. Zhang, and L. Chen, "Recent advances of single-object tracking methods: A brief survey," *Neurocomputing*, vol. 455, pp. 1–11, 2021.

[13] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *arXiv preprint arXiv:2012.13392*, 2020.

[14] T. D. Nguyen and M. Kresovic, "A survey of top-down approaches for human pose estimation," *arXiv preprint arXiv:2202.02656*, 2022.

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

[16] K. Chen, "Sitting posture recognition based on openpose," in *IOP Conference Series: Materials Science and Engineering*, vol. 677, no. 3. IOP Publishing, 2019, p. 032057.

[17] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5386–5395.

[18] B. Jo and S. Kim, "Comparative analysis of openpose, posenet, and movenet models for pose estimation in mobile devices." *Traitement du Signal*, vol. 39, no. 1, pp. 119–124, 2022.

[19] P.-L. Liu and C.-C. Chang, "Simple method integrating openpose and rgb-d camera for identifying 3d body landmark locations in various postures," *International Journal of Industrial Ergonomics*, vol. 91, p. 103354, 2022.

[20] K. Lee, Y. Ban, and S. Lee, "Efficient depth enhancement using a combination of color and depth information," *Sensors*, vol. 17, no. 7, p. 1544, 2017.

[21] X. Zhang and R. Wu, "Fast depth image denoising and enhancement using a deep convolutional network," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2499–2503.

[22] L. Needham, M. Evans, D. P. Cosker, L. Wade, P. M. McGuigan, J. L. Bilzon, and S. L. Colyer, "The accuracy of several pose estimation methods for 3d joint centre localisation," *Scientific reports*, vol. 11, no. 1, p. 20673, 2021.

[23] M. M. Ibrahim, Q. Liu, R. Khan, J. Yang, E. Adeli, and Y. Yang, "Depth map artefacts reduction: A review," *IET Image Processing*, vol. 14, no. 12, pp. 2630–2644, 2020.

[24] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "Ghum & ghuml: Generative 3d human shape and articulated pose models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6184–6193.

[25] S. Mroz, N. Baddour, C. McGuirk, P. Juneau, A. Tu, K. Cheung, and E. Lemaire, "Comparing the quality of human pose estimation with blazepose or openpose," in *2021 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART)*. IEEE, 2021, pp. 1–4.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields. arxiv 2018," *arXiv preprint arXiv:1812.08008*, 1812.

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

[29] ——, "Z3: An efficient smt solver," in *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2008, pp. 337–340.

[30] E. Woo, P. White, and C. Lai, "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] "Taekwondo student manual," https://taekwondoamerica.org/wp-content/uploads/2017/09/Student-Manual-2012.pdf, accessed: 20 December 2023.
[32] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. J. Király, "sktime: A unified interface for machine learning with time series," *arXiv preprint arXiv:1909.07872*, 2019.
[33] H. Cho, H.-J. Choi, C.-E. Lee, and C.-W. Sir, "Sitting posture prediction and correction system using arduino-based chair and deep learning model," in *2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)*. IEEE, 2019, pp. 98–102.
[34] R. Gupta, D. Saini, and S. Mishra, "Posture detection using deep learning for time series data," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, 2020, pp. 740–744.
[35] G. Liu, L. Lin, W. Zhou, R. Zhang, H. Yin, J. Chen, and H. Guo, "A posture recognition method applied to smart product service," *Procedia CIRP*, vol. 83, pp. 425–428, 2019.