A Physiotherapy Video Matching Method Supporting Arbitrary Camera Placement via Angle-of-Limb-based Posture Structures

***Abstract*—"Hospital at home" enlightens revolutions of medical service automation by leading in modern information technologies. This paper revisited the remote physiotherapy service, considering that convalescents at home can record their physiotherapy exercises with an arbitrarily-placed mobile device. In this paper, we proposed a physiotherapy video matching method supporting arbitrary camera placement so that physiotherapy movements in video clips of free camera shoot angles can be accurately matched. We formulate the physiotherapy video matching problem from the optimization perspective and solve it with a pipeline consisting of certain single objective modules. To alleviate the effect of different camera shooting angles between clips of mentors and convalescents, the angle-of-limb-based posture structure (ALPS) and associated camera-angle-free (CAFE) transformation are invented for representing physiotherapy movements. Moreover, the three-phase ALPS matching algorithm (TALMA) is developed to to find out the best movement matching between physiotherapy video clips. Real-world experiments in physiotherapy scenarios are conducted to show the effectiveness of the proposed method. Experimental results reveal that our proposed method indeed achieves superior performance in precision and practicality. The real-world tests also show that the time difference between our results and the ground truths is less than 0.07 seconds, almost indistinguishable from human experts. The developed prototype, test datasets, and demo video are available at: https://github.com/NCKU-CIoTlab/TALMA-on-ALPS/.**

keywords: rehabilitation modeling, applied deep learning, telehealthcare automation, semantic matching, hospital at home.

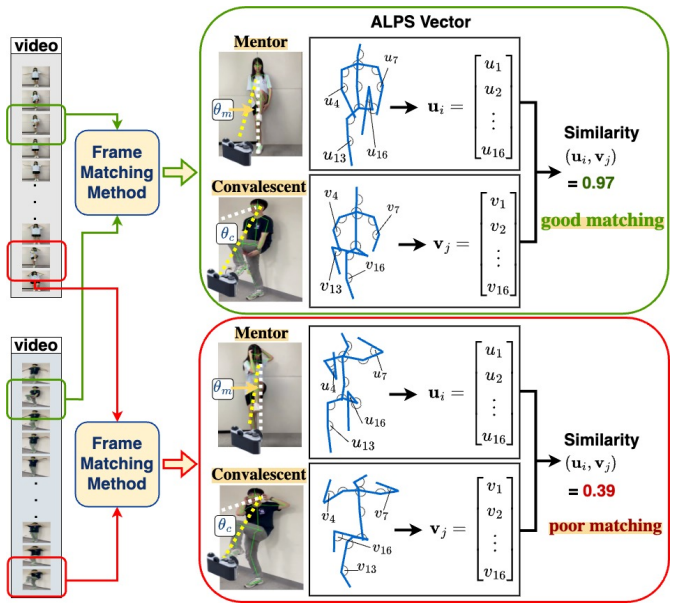


Fig. 1: Illustrating physiotherapy video matching scenarios by using our proposed ALPS-based matching. Note that the two clips are shot from different camera angles, where the camera is placed in front of the mentor but placed on the left side of the convalescent. Our method alleviates the camera-angle effect and appearance characteristics to produce matching results.

I. Introduction

"Hospital at home" is a widely promoted healthcare policy recently and is urgently required in countries with declining birthrate and aging populations [1], [2]. Healthcare research in robotic and automation societies drives core technologies that enable the development of remote hospital applications. Some departments whose therapy processes can be analyzed via computer vision technologies, such as physiotherapy [3], [4], [5], [6], [7], are considered the first wave to advocate the hospital-at-home realm. Thus, this paper adopts physiotherapy to investigate issues of leading IT and automation technologies to hospital-at-home industries.

In developing hospital-at-home healthcare applications, the latest IT and AI technologies need to be integrated deeply. Fig. 1 shows a remote physiotherapy application that automatically finds movements matching from two video clips (or clips, for short): one from a female mentor and the other from a male convalescent. The physiotherapy video matching problem is to find out frames from the convalescent's clip which are accordingly most similar to the selected frames (called anchor frames) in the mentor's clip, so that the medical experts can rapidly review the uploaded convalescent clips to significantly reduce time costs in frame-by-frame or second-by-second verification. This solution not only reduces working stress for medical personnel but also increases service capacity to serve more at-home convalescents.

Two technical difficulties need to be tackled in the physiotherapy video matching problem. The first is to understand physiotherapy movement semantics in different camera shooting angles (or shortly, camera angles). This difficulty comes from the property of remote physiotherapy applications, where at-home convalescents may record their clip by placing their camera devices in a shooting angle different from mentor's. Fig. 1 shows an example of comparing two frame pairs to demonstrate the difficulty: the upper block is a similar case, and the lower is dissimilar. One is very possibly hard to discriminate between similar/dissimilar cases by merely viewing frames, whereas the 3D models (by our method) give more cues to distinguish the two cases. The task becomes harder in long video clips with very different camera shooting angles. The second is to effectively find a proper subset from a large set of frames by considering the physiotherapy movement semantics. The essence of this difficulty inheres the computational matching problem with additional physiotherapy restrictions, such as no multiple convalescent frames matching to a mentor's frame and unstable similarity comparison due to different

camera angles. These difficulties make the physiotherapy video matching problem a complicated research task.

Physiotherapy movement analysis has been widely studied, with many approaches assessing postural stability and joint motion using static and predefined angles. Existing works, such as Q-angle evaluations [8] and single-plane observations [3], demonstrated reliability under fixed viewpoints but failed to address camera angle variations, which is a challenge arising from camera placement in remote physiotherapy setups. Efforts of improving accuracy, including software-assisted assessments [4] and motion analysis systems [9], provided stable results but relied on consistent perspectives. Similarly, tools such as Kinect sensors [5] maintained the measurement stability but remained constrained by static viewpoints. Frameworks for movement variability [10] and postural assessments [11] gave insights into static conditions but lacked mechanisms to handle viewpoint shifts or enable customizable evaluation criteria. To the best of our survey, no prior work address camera-angle variations or support dynamic assessment criteria, making comparisons with existing methods infeasible. This gap motivates us to tackle the camera-angle difficulties, advancing physiotherapy assessments beyond the limitations of prior approaches.

This paper addresses the above technical challenges and proposes a physiotherapy movement matching (PVM) method to match the convalescent clip to the mentor's. Our proposed method is founded upon three innovative designs. Firstly, we deconstruct the PVM problem by an optimization analysis with abstract functions. The PVM solution turns into designing a pipeline and implementing each function to fulfill the optimization analysis so that certain single objective modules work together to solve complex and entangled difficulties in the PVM problem. In addition, functions of extracting keypoints from frames are implemented by leveraging existing pretrained deepnet models for rapid development. Secondly, we invent an angle-of-limb-based posture structure (ALPS) to preserve most posture properties for physiotherapy semantics in different camera angles. The camera-angle-free (CAFE) transformation is devised to systematically transform the keypoint models obtained by deepnets to the ALPS models. The ALPS plays the fundamental data representation in the following physiotherapy video matching algorithm. Thirdly, we design a three-phase ALPS matching algorithm (TALMA) for achieving accurate and robust physiotherapy video matching. As the matching problem is a well-known NP-complete problem [12], the TALMA is thus developed to achieve sub-optimum solutions in a heuristic manner of considering physiotherapy semantics and matching quality. To validate our proposed method, we conduct real-world experiments on the physiotherapy scenarios. The mentor player, who has a national nurse license in Taiwan, performs 11 exemplary physiotherapy movements that are widely used in related industries to support the practicability of the experimental results. The results reveal that our proposed method indeed achieves superior matching performance, and the mean time difference of results between our method and the ground truths is less than 0.07 seconds in the convalescent clip, almost indistinguishable from human experts. The demo video of experimental results and codes[1] are available at https://github.com/NCKU-CIoTlab/TALMA-on-ALPS/.

II. Problem Formulation and Sketch of Solution

For ease of understanding, we define a symbol convention for a human posture vector (e.g., $\mathbf{h}$)[2] throughout this paper as follows, where the environmental factors (camera angle[3] and owner role of data) and the semantic factors (frame index and element index) are shown simultaneously:

$$\begin{matrix}\theta\text{: camera angle} \\ r\text{: owner role of } \mathbf{h}\end{matrix} \quad {}^{\theta}_{r}\mathbf{h}^{i}_{j} \quad \begin{matrix}i\text{: frame index} \\ j\text{: vector element index}\end{matrix}$$

The owner role r could be a mentor or a convalescent, indicated by m and c, respectively. Fig. 2 illustrates a mapping instance between the symbol definition and a physiotherapy video frame.

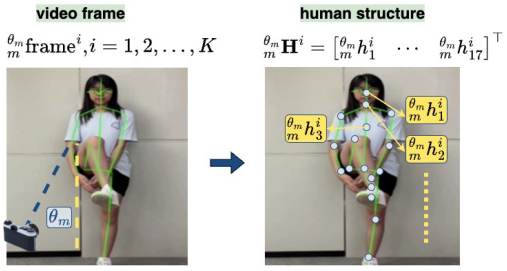


Fig. 2: The mapping between the symbol definition and a physiotherapy video frame.

A video clip consists of a set of ordered frames. For a mentor, the ordered frame set is denoted by ${}^{\theta_m}_{m}\mathcal{V} = \{{}^{\theta_m}_{m}\text{frame}^i | i = 1, \ldots, M\}$, where ${}^{\theta_m}_{m}\text{frame}^i$ represents the i-th frame of the mentor (m) with the camera angle θ_m. Similarly, the ordered frame set, recorded by a convalescent, is represented as ${}^{\theta_c}_{c}\mathcal{V} = \{{}^{\theta_c}_{c}\text{frame}^i | i = 1, \ldots, N\}$, where ${}^{\theta_c}_{c}\text{frame}^i$ represents the i-th frame of the convalescent with the camera angle θ_c. The mentor can designate K anchor frames in ${}^{\theta_m}_{m}\mathcal{V}$, whose index are denoted by α_i where $i = 1, \ldots, K$ and $1 \leq \alpha_i \leq N$, where an anchor frame is a referential standard that the mentor asks convalescents to perform the same movement in ${}^{\theta_c}_{c}\mathcal{V}$ to meet physiotherapy requirements. A fundamental function in remote physiotherapy services is the physiotherapy video matching (PVM), which selects a subset of ${}^{\theta_c}_{c}\mathcal{V}$ to optimally match $\{{}^{\theta_m}_{m}\text{frame}^{\alpha_i} | i = 1, \ldots, K\}$. With such capability, hospital physiotherapists can automatically assess rehabilitation progress for a large number of convalescents in a short time. We formulate the PVM problem as follows.

Problem 1 (Physiotherapy Video Matching (PVM))**.** *Let $\alpha_i, i = 1, \ldots, K$, be the index of the anchor frames. Find K frame index β_j from ${}^{\theta_c}_{c}\mathcal{V}$, denoted by $\hat{Z}^* = \{\beta_j | j = 1, \ldots, K$ and $1 \leq \beta_j \leq N\}$, such that the selected K frames are most similar to the corresponding anchor frames in terms*

[1]Due to the intellectual property issue, source codes will be released after acceptance.

[2]The usage of superscripts and subscripts will keep consistency in other symbols of similar purposes.

[3]The camera angle in this paper means the angle of a person in the viewpoint of the camera, which is affected by the camera placement.

of movement measures under the following requirements in remote physiotherapy scenarios:

1) *the physiotherapy movement comparison between frames shall avoid interference from visual appearance characteristics, such as weight, height, clothes, etc., and*
2) *the physiotherapy video clips for the mentor and convalescents can be shot in arbitrarily camera angles.*

For computationally studying the issue, the problem can be represented as an optimization form:

$$\textit{Maximize} \underbrace{sim\left(\left\{{}^{\theta_m}_{m}\textit{frame}^{\alpha_i}\right\}_{i=1}^{K}, \left\{{}^{\theta_c}_{c}\textit{frame}^{\beta_j}\right\}_{j=1}^{K}\right)}_{\textbf{Need to satisfy requirements (1) and (2).}} \quad (1)$$

The solution $\hat{Z}^*$ to Eq.(1) is thus expressed in the equation:

$$\hat{Z}^* = \arg\max_{\{\beta_i\}} \sum_{i=1}^{K} \textit{sim}({}^{\theta_m}_{m}\textit{frame}^{\alpha_i}, {}^{\theta_c}_{c}\textit{frame}^{\beta_i}) \quad (2)$$

■

Obviously, finding out the solution $\hat{Z}^*$ satisfying the above remote physiotherapy requirements in Problem 1 from directly comparing similarity between video clips ${}^{\theta_m}_{m}\mathcal{V}$ and ${}^{\theta_c}_{c}\mathcal{V}$ is a challenging and complicated task, as the PVM problem is essentially the classic NP-complete problem [12] with additional domain-dependent constraints. Inspired by some classic theorems, such as Fourier Transform, that avoids time-domain signal processing of high computational complexity by using frequency-domain processing of low complexity, we tackle the above PVM problem with a similar trick: *we transform the video data to a camera-angle-free domain and then perform physiotherapy movement matching on that domain*. Assume that the ideal data model ${}_{r}\mathcal{A}^i$ for role r ($r \in \{m, c\}$) of frame i provides a unified appearance-independent and camera-angle-free representation for human bodies. Problem 1 can thus be expressed in the following form:

$$\hat{Z}^* = \arg\max_{\{\beta_i\}} \sum_{i=1}^{K} \underbrace{\text{sim}({}_{m}\mathcal{A}^{\alpha_i}, {}_{c}\mathcal{A}^{\beta_i})}_{\textbf{(1) appearance-independent structure, (2) no camera angles.}} \quad (3)$$

Note that the two requirements in Problem 1 are considered in designing the data model ${}_{r}\mathcal{A}^i$. Then, for bridging Eq.(2) and Eq.(3), we invent three abstract functions[4] as solution representation tools to construct ${}_{r}\mathcal{A}^i$. The first abstract function $\mathfrak{F}_{\text{PHP}}$ is to extract a two-dimensional (2D) keypoint-based structure called the purified human posture (PHP) model, denoted by ${}^{\theta_r}_{r}\mathbf{H}^i$ for the i-th video frame of role r with the camera angle θ_r, and is defined as:

$${}^{\theta_r}_{r}\mathbf{H}^i \triangleq \mathfrak{F}_{\text{PHP}}({}^{\theta_r}_{r}\text{frame}^i) \quad (4)$$

The second abstract function $\mathfrak{F}_{\text{3DPHP}}$ is to up-project a 2D PHP model ${}^{\theta_r}_{r}\mathbf{H}^i$ to a 3D PHP model ${}^{\theta_r}_{r}\mathbf{Q}^i$, defined as:

$${}^{\theta_r}_{r}\mathbf{Q}^i \triangleq \mathfrak{F}_{\text{3DPHP}}({}^{\theta_r}_{r}\mathbf{H}^i) \quad (5)$$

The 3D PHP structure ${}^{\theta_r}_{r}\mathbf{Q}^i$ provides the capability to handle the camera-angle effect. The third abstract function $\mathfrak{F}_{\text{CAFE}}$ is to generate a camera-angle-free structure ${}_{r}\mathbf{A}^i$ (no camera angle parameter) for a 3D PHP structure ${}^{\theta_r}_{r}\mathbf{Q}^i$, defined as:

$${}_{r}\mathbf{A}^i \triangleq \mathfrak{F}_{\text{CAFE}}({}^{\theta_r}_{r}\mathbf{Q}^i) \quad (6)$$

Note that ${}_{r}\mathbf{A}^i$ is the implementation of the idea model ${}_{r}\mathcal{A}^i$.

With the composition of the above three abstract functions, the PVM solution form in Eq.(2) can be derived as follows:

$$\begin{aligned}\hat{Z}^* &= \arg\max_{\{\beta_i\}} \sum_{i=1}^{K} \text{sim}({}^{\theta_m}_{m}\text{frame}^{\alpha_i}, {}^{\theta_c}_{c}\text{frame}^{\beta_i}) \\ &= \arg\max_{\{\beta_i\}} \sum_{i=1}^{K} \text{sim}(\mathfrak{F}_{\text{CAFE}} \circ \mathfrak{F}_{\text{3DPHP}} \circ \mathfrak{F}_{\text{PHP}}({}^{\theta_m}_{m}\text{frame}^{\alpha_i}), \\ &\qquad \mathfrak{F}_{\text{CAFE}} \circ \mathfrak{F}_{\text{3DPHP}} \circ \mathfrak{F}_{\text{PHP}}({}^{\theta_c}_{c}\text{frame}^{\beta_i})) \quad (7)\end{aligned}$$

(Processing frames by 3 abstract functions without changing optimization goal.)

$$= \arg\max_{\{\beta_i\}} \sum_{i=1}^{K} \text{sim}({}_{m}\mathbf{A}^{\alpha_i}, {}_{c}\mathbf{A}^{\beta_i}) \quad \text{(connecting Eq. (3) to Eq.(2))}$$

$$= \underbrace{\arg\min_{\{\beta_i\}} \sum_{i=1}^{K} 1 - \cos({}_{m}\mathbf{A}^{\alpha_i}, {}_{c}\mathbf{A}^{\beta_i})}_{\text{A combinatorial optimization problem.}} \quad (8)$$

("$\text{sim}({}_{m}\mathbf{A}^{\alpha_i}, {}_{c}\mathbf{A}^{\beta_i})\uparrow$" is equivalent to "$\cos({}_{m}\mathbf{A}^{\alpha_i}, {}_{c}\mathbf{A}^{\beta_i}) \to 1$")

Eq. (8) exposes our solution sketch: *the PVM solution $\hat{Z}^*$ can be found out from the combinatorial optimization perspective in case that the three abstract functions (Eqs.(4-6)) are realized (referring to Eq. (7))*. The following sections give the complete technical details of implementing the solution sketch.

III. Proposed Pipeline for Solving PVM

Based on Eqs.(7-8), Fig. 3 illustrates an overall pipeline for solving Problem 1. The pipeline is composed of four modules: the first three, i.e., PHP Net, 3DPHP Net, and CAFE Transformation, carry out our invented abstract functions, including $\mathfrak{F}_{\text{PHP}}$, $\mathfrak{F}_{\text{3DPHP}}$, and $\mathfrak{F}_{\text{CAFE}}$ (referring to Eq. (7)), and the last, i.e., TALMA, solves the optimization problem of Eq.(8). The processing flow of the pipeline is merely to perform the four components in sequence, which is consistent to the function execution order in Eq.(7).

The proposed pipeline can be divided into three stages to look into its processing philosophy: extraction, transformation, and matching. The extraction stage contains the PHP Net, which implements Eq. (4) and extracts human keypoints ${}^{\theta_r}_{r}\mathbf{H}^i$ from video clips for represent posture semantics. The transformation stage contains the 3DPHP Net and the camera-angle-free (CAFE) transformation, which generate 3D human keypoints ${}^{\theta_r}_{r}\mathbf{Q}^i$ for implementing Eq. (5), and the angle-of-limb-based posture structure (ALPS) ${}_{r}\mathbf{A}^i$ (will be discussed later) for implementing Eq. (6), respectively, in order to represent human movements that meet remote physiotherapy requirements. The matching stage contains the three-phase ALPS matching algorithm (TALMA), which produces the matching result $\hat{Z}^*$ for the PVM problem of Eq.(8). ALL the technical details will be presented in Sec. IV.

For satisfying requirement (1) in in Problem 1 that video matching between a mentor and a convalescent shall not be

[4]We call abstract functions as only the function forms are presented here for explaining the high-level design philosophy for our solution, and their implementation details will be presented in the next section.

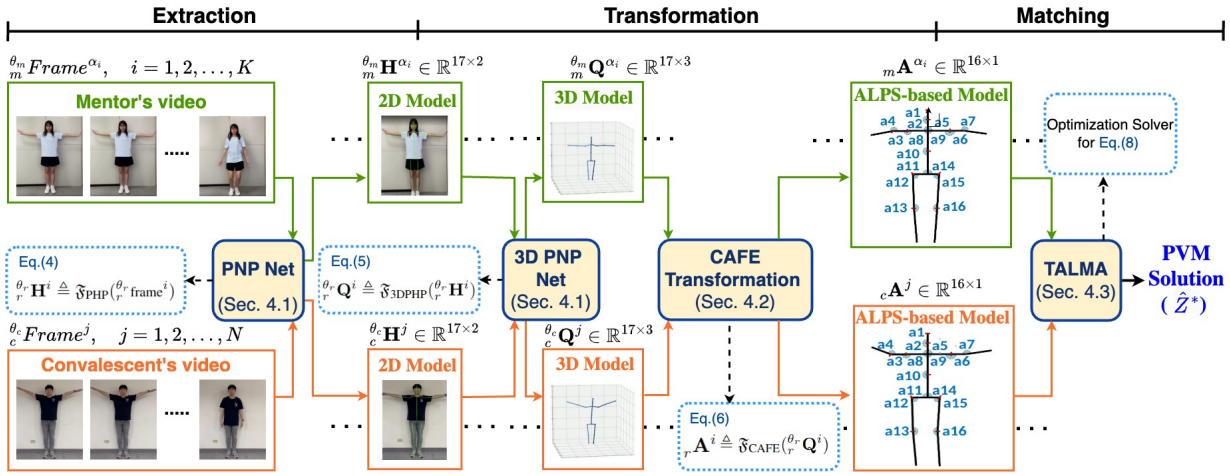


Fig. 3: The proposed PVM pipeline for computing solution sketch in Eqs. (7-8).

affected by visual appearance features, we thus adopt the pose-relevant features in this work. Fig. 4 shows the definitions of human structures used in this work, and the mathematical definitions of those used in Fig. 3 are described as follows. The two-dimensional PHP model ${}^{\theta_r}_{r}\mathbf{H}^i$ (shown in Fig. 4(a)) based on the keypoint-based model is defined as:

$$ {}^{\theta_r}_{r}\mathbf{H}^i \triangleq \begin{bmatrix} {}^{\theta_r}_{r}\mathbf{h}^i_1 & \cdots & {}^{\theta_r}_{r}\mathbf{h}^i_{17} \end{bmatrix}^\top \in \mathbb{R}^{17\times 2}, \quad \text{where } {}^{\theta_r}_{r}\mathbf{h}^i_j = \begin{bmatrix} x_j & y_j \end{bmatrix} \in \mathbb{R}^{1\times 2},\ j = 1,\ldots,17. \tag{9} $$

$[x_j\ y_j]$ is the 2D coordinate of keypoint ${}^{\theta_r}_{r}\mathbf{h}^i_j$. The 3D PHP model ${}^{\theta_r}_{r}\mathbf{Q}^j$ follows the same structure definition in Fig. 4(a), except keypoints are raised to 3D vectors, defined as:

$$ {}^{\theta_r}_{r}\mathbf{Q}^i \triangleq \begin{bmatrix} {}^{\theta_r}_{r}\mathbf{q}^i_1 & \cdots & {}^{\theta_r}_{r}\mathbf{q}^i_{17} \end{bmatrix}^\top \in \mathbb{R}^{17\times 3}, \quad \text{where } {}^{\theta_r}_{r}\mathbf{q}^i_j = \begin{bmatrix} x_j & y_j & z_j \end{bmatrix} \in \mathbb{R}^{1\times 3}\ j = 1,\ldots,17. \tag{10} $$

Note that under different camera angles, a human posture in the keypoint-based model (e.g., ${}^{\theta_r}_{r}\mathbf{H}^i$ and ${}^{\theta_r}_{r}\mathbf{Q}^i$) could be valued very differently, which brings out difficulties in computing posture similarity from video clips, such as the frame discriminating examples in Fig. 1. Thus, the angle-of-limb-based posture structure (ALPS), denoted by ${}_{r}\mathbf{A}^i$, for frame i of role r is designed to use fewer variant posture features to model human movements to cope with the camera-angle effect and preserve the most movement semantics for satisfying the two requirements in Problem 1. The ALPS model ${}_{r}\mathbf{A}^i$, shown in Fig. 4(c), is defined as:

$$ {}_{r}\mathbf{A}^i \triangleq \begin{bmatrix} {}_{r}a^i_1 & \cdots & {}_{r}a^i_{16} \end{bmatrix}^\top \in \mathbb{R}^{16\times 1} \tag{11} $$

where ${}_{r}a^i_j$ represents the angle of two adjacent limbs and its details will be presented in Sec. IV-B.

IV. Key Component Designs

A. Deepnet-empowered Function Implementation: PHP-Net and 3DPHP-Net

It is unnecessary to re-invent all wheels to solve the PVM problem. For carrying out the abstract functions $\mathfrak{F}_{\text{PHP}}$ and $\mathfrak{F}_{\text{3DPHP}}$, the first two components, PHP-Net and 3DPHP-Net, in the pipeline employ existing deep neural network (deepnet) modules, Alphapose [6] and Dual-stream Spatio-temporal Transformer (DST) [7], respectively, in the human pose research area. Thus, ${}^{\theta_r}_{r}\mathbf{H}^i$ and ${}^{\theta_r}_{r}\mathbf{Q}^i$ can be swiftly obtained by utilizing the deepnet models, represented as the follows:

$$ {}^{\theta_r}_{r}\mathbf{H}^i \triangleq \mathfrak{F}_{\text{PHP}}({}^{\theta_r}_{r}\text{frame}^i) = \text{Alphapose}({}^{\theta_r}_{r}\text{frame}^i) \tag{12} $$

$$ {}^{\theta_r}_{r}\mathbf{Q}^i \triangleq \mathfrak{F}_{\text{3DPHP}}({}^{\theta_r}_{r}\mathbf{H}^i) = \text{DST}({}^{\theta_r}_{r}\mathbf{H}^i) \tag{13} $$

In this work, we only format the results of Alphapose and DST to fit the keypoints defined in Fig. 4(a) and then generates ${}^{\theta_r}_{r}\mathbf{H}^i$ and ${}^{\theta_r}_{r}\mathbf{Q}^i$ for later processing.

B. Angle-of-Limb-based Posture Structure (ALPS) and CAFE Transformation

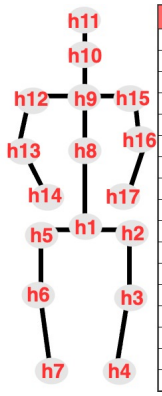
Since the keypoint-based features are highly related to the human appearance, we use the angle-of-limb vectors to describe human movements. The idea comes from that angles of limbs are mostly invariant in comparing two human representations of different camera angles. The 3D keypoint model ${}^{\theta_r}_{r}\mathbf{Q}^i \in \mathbb{R}^{17\times 3}$ is transformed into the limb-based model ${}^{\theta_r}_{r}\mathbf{L}^i \in \mathbb{R}^{16\times 3}$ by connecting the corresponding keypoint pairs shown in Fig. 4(b), defined as:

$$ {}^{\theta_r}_{r}\mathbf{L}^i \triangleq \begin{bmatrix} {}^{\theta_r}_{r}\mathbf{l}^i_1 & \cdots & {}^{\theta_r}_{r}\mathbf{l}^i_{16} \end{bmatrix}^\top \in \mathbb{R}^{16\times 3}, \text{ where } {}^{\theta_r}_{r}\mathbf{l}^i_j = {}^{\theta_r}_{r}\mathbf{q}^i_t - {}^{\theta_r}_{r}\mathbf{q}^i_s = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} \in \mathbb{R}^{1\times 3},\ j = 1,\ldots,16. \tag{14} $$

Note that ${}^{\theta_r}_{r}\mathbf{q}^i_t - {}^{\theta_r}_{r}\mathbf{q}^i_s$ represents the j-th limb vector defined in Fig. 4(b). Also notice that the process of transforming ${}^{\theta_r}_{r}\mathbf{Q}^i$ into ${}^{\theta_r}_{r}\mathbf{L}^i$ can be accomplished by simply a matrix multiplication:

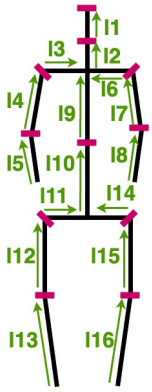
$$ {}^{\theta_r}_{r}\mathbf{L}^i = \mathbf{W}^{\mathbf{L}} \times {}^{\theta_r}_{r}\mathbf{Q}^i = \begin{bmatrix} {}^{\theta_r}_{r}\mathbf{l}^i_1 & \cdots & {}^{\theta_r}_{r}\mathbf{l}^i_{16} \end{bmatrix}^\top \in \mathbb{R}^{16\times 3} \tag{15} $$

where $\mathbf{W}^{\mathbf{L}} \in \mathbb{R}^{16\times 17}$ is a limb-description matrix. The j-th row of $\mathbf{W}^{\mathbf{L}}$ describes the start and end keypoints of the j-th limb, filling with -1 and 1 in the s-th and t-th elements, respectively; others are 0. In the structure definition of Fig. 4(b), $\mathbf{W}^{\mathbf{L}}$ is set as follows:



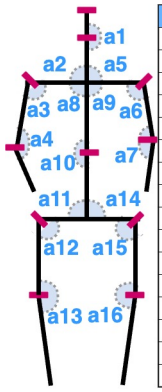
ID	Keypoint
h1	Root
h2	Right Hip
h3	Right Knee
h4	Right Ankle
h5	Left Hip
h6	Left Knee
h7	Left Ankle
h8	Torse
h9	Neck
h10	Nose
h11	Head
h12	Left Shoulder
h13	Left Elbow
h14	Left Wrist
h15	Right Shoulder
h16	Right Elbow
h17	Right Wrist

(a) The keypoint-based human structure



ID	Limb	Formula
l1	Head	q11-q10
l2	Neck	q10-q9
l3	Right Shoulder	q9-q12
l4	Right Upperarm	q12-q13
l5	Right Forearm	q13-q14
l6	Left Shoulder	q9-q15
l7	Left Upperarm	q15-q16
l8	Left Forearm	q16-q17
l9	Chest	q9-q8
l10	Stomach	q8-q1
l11	Right Hip	q1-q5
l12	Right Thigh	q5-q6
l13	Right Shank	q6-q7
l14	Left Hip	q1-q2
l15	Left Thigh	q2-q3
l16	Left Shank	q3-q4

(b) The limb-based structures



ID	Limb Pair for Angle	Formula
a1	(Head, Neck)	cos(l1, l2)
a2	(Neck, Right Shoulder)	cos(l2, l3)
a3	(Right Shoulder, Right Upperarm)	cos(l4, l3)
a4	(Right Upperarm, Right Forearm)	cos(l5, l4)
a5	(Neck, Left Shoulder)	cos(l2, l6)
a6	(Left Shoulder, Left Upperarm)	cos(l7, l6)
a7	(Left Upperarm, Left Forearm)	cos(l8, l7)
a8	(Right Shoulder, Chest)	cos(l3, l9)
a9	(Left Shoulder, Chest)	cos(l6, l9)
a10	(Chest, Stomach)	cos(l10, l9)
a11	(Stomach, Right Hip)	cos(l11, l10)
a12	(Right Hip, Right Thigh)	cos(l12, l11)
a13	(Right Thigh, Right Shank)	cos(l13, l12)
a14	(Stomach, Left Hip)	cos(l10, l14)
a15	(Left Hip, Left Thigh)	cos(l15, l14)
a16	(Left Thigh, Left Shank)	cos(l16, l15)

(c) The angle-of-limb-based posture structures

Fig. 4: Structure definitions used in this work: (a) is for $\mathbf{H}$ and $\mathbf{Q}$, where $\mathbf{Q}$ is the 3D version of $\mathbf{H}$, (b) is for $\mathbf{L}$, (c) is for $\mathbf{A}$.

$$\mathbf{W}^{\mathbf{L}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

The angle-of-limb-based posture structure (ALPS) ${}_r\mathbf{A}^i \in \mathbb{R}^{16\times 1}$ is a processed vector from ${}^{\theta_r}_{r}\mathbf{L}^i$, where each component describes the angle of two connected limbs, computed by their cosine value. The predefined limb pairs used in this work are shown in Fig. 4(c). The ALPS model ${}_r\mathbf{A}^i$ corresponding to ${}^{\theta_r}_{r}\mathbf{L}^i$ can be derived below. Let $\mathbf{U} \in \mathbb{R}^{16\times 3}$ be a unit limb-vector collection of ${}^{\theta_r}_{r}\mathbf{L}^i$. That is, the j-th row of $\mathbf{U}$, expressed by $\text{Row}_j(\mathbf{U})$, can be written as:

$$\text{Row}_j(\mathbf{U}) = {}^{\theta_r}_{r}\mathbf{l}^i_j / \|{}^{\theta_r}_{r}\mathbf{l}^i_j\|_2 \in \mathbb{R}^{1\times 3},\ j = 1, \dots 16. \tag{17}$$

Assume ${}_ra^i_j$, the j-th element of ${}_r\mathbf{A}^i$, is the angle between the s-th and e-th limbs, that is, ${}_ra^i_j = \cos(s\text{-th limb}, e\text{-th limb})$. Let $\mathbf{W}^{\mathbf{A}}_S$ and $\mathbf{W}^{\mathbf{A}}_E$ of $\mathbb{R}^{16\times 16}$ be limb-connectivity matrices for computing ${}_ra^i_j$. For the j-th connected limb pair $(\mathbf{l}_s, \mathbf{l}_e)$ of angle ${}_ra^i_j$, elements $\mathbf{W}^{\mathbf{A}}_S[j][s]$ and $\mathbf{W}^{\mathbf{A}}_E[j][e]$ are set to 1, respectively; other elements in the j-th row are 0. In the definition of Fig. 4(c), $\mathbf{W}^{\mathbf{A}}_S$ and $\mathbf{W}^{\mathbf{A}}_E$ are thus set as follows:

$$\mathbf{W}^{\mathbf{A}}_S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{W}^{\mathbf{A}}_E = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{18}$$

Then, ${}_ra^i_j$, the j-th component of ${}_r\mathbf{A}^i$, equals to the inner product of unit vectors for the j-th limb pair, expressed as:

$$\begin{aligned} {}_ra^i_j &= \left(\text{Row}_j\left(\mathbf{W}^{\mathbf{A}}_S\right)\cdot\mathbf{U}\right)\cdot\left(\text{Row}_j\left(\mathbf{W}^{\mathbf{A}}_E\right)\cdot\mathbf{U}\right)^\top \\ &= \underbrace{\left(\text{Row}_s(\mathbf{U})\right)}_{\text{row vector}}\cdot\underbrace{\left(\text{Row}_e(\mathbf{U})\right)^\top}_{\text{column vector}} = \cos\left({}^{\theta_r}_{r}\mathbf{l}^i_s, {}^{\theta_r}_{r}\mathbf{l}^i_e\right) \end{aligned} \tag{19}$$

By Eq.(19), the ALPS model ${}_r\mathbf{A}^i$ can be computed by merely using $\mathbf{W}^{\mathbf{A}}_S$, $\mathbf{W}^{\mathbf{A}}_E$, and $\mathbf{U}$ in the following equation:

$${}_r\mathbf{A}^i = \left[\left(\text{Row}_j\left(\mathbf{W}^{\mathbf{A}}_S\right)\cdot\mathbf{U}\right)\cdot\left(\text{Row}_j\left(\mathbf{W}^{\mathbf{A}}_E\right)\cdot\mathbf{U}\right)^\top\right]_{j=1}^{16} \tag{20}$$

Taking Eqs. (19-20) to Eq. (11) obtains the computational expression of the CAFE transformation (i.e., $\mathfrak{F}_{\text{CAFE}}$):

$${}_r\mathbf{A}^i \triangleq [{}_ra^i_1, \cdots, {}_ra^i_{16}]^\top = \left[\cos\left({}^{\theta_r}_{r}\mathbf{l}^i_s, {}^{\theta_r}_{r}\mathbf{l}^i_e\right) \text{ of Row } j\right]_{j=1}^{16} \tag{21}$$

Since Eq. (14) connects ${}^{\theta_r}_{r}\mathbf{Q}^i$ to ${}^{\theta_r}_{r}\mathbf{L}^i$, the model ${}^{\theta_r}_{r}\mathbf{Q}^i = \mathfrak{F}_{\text{3DPHP}} \circ \mathfrak{F}_{\text{PHP}}({}^{\theta_r}_{r}\text{frame}^i)$ obtained by deepnets can be successfully transformed into the ALPS model ${}_r\mathbf{A}^i$ by Eq. (21).

C. Three-Phase ALPS Matching Algorithm for PVM Solver

Duplicating ALPS for Matching Robustness. To reduce interference from hidden angle information due to different camera angles, we use full-body, left-body, and right-body ALPS models in the following matching algorithm. The full-body ALPS, denoted by ${}^{F}_{r}\mathbf{A}^i$, of frame i equals ${}_r\mathbf{A}^i$. The left-body ALPS, denoted by ${}^{L}_{r}\mathbf{A}^i$, is the ${}_r\mathbf{A}^i$ with setting elements

in the right-side body to 0, i.e., $a_2 = a_3 = a_4 = a_8 = a_{11} = a_{12} = a_{13} = 0$. Similarly, the right-body ALPS, denoted by ${}^{R}_{r}\mathbf{A}^i$, is the ${}_{r}\mathbf{A}^i$ with setting elements in the left-side body to 0, i.e., $a_5 = a_6 = a_7 = a_9 = a_{14} = a_{15} = a_{16} = 0$.

Algorithm 1 depicts our proposed Three-phase ALPS Matching Algorithm (TALMA) for solving the PVM problem with ALPS models corresponding clips of the mentor and a convalescent and the K specified anchor frames, i.e., $\alpha_1, \ldots, \alpha_K$. The design idea is prompted by a general machining procedure [13]: sequentially performing a rough process, a deep process, and a finishing process on workpieces to produce final products, which can be analogous to three phases of our proposed TALMA: a rough-matching phase, a fine-matching phase, and an integration phase for repeatedly working out a finishing frame-matching set in different context levels on ALPS models. Algorithm 1 gives the detailed computational steps for reproducibility, and we describe the associated conceptual explanations for readability due to the length limit. The first phase (Lines 9-21), called the rough-matching phase, is to find the matching frames to each anchor frame α_i, $i = 1, \ldots, K$, in three ALPS models by leveraging the dynamic time warping (DTW) [14], a widely used computational matching method. The Phase-1 results, denoted by ${}^{[F|L|R]}\mathbf{P1R}$[5], contains the matching frame set generated by the DTW on three ALPS models. Note that Step 1.2 based on the DTW inherently incurs that many convalescent frames match to one anchor frame (i.e., a many-to-one matching). This rough-matching phase is named as its results do not fit the physiotherapy scenarios. The second phase, called the fine-matching phase, (Lines 22-48) is thus designed to transform the many-to-one matching to the one-to-one matching according to the full-body similarity (Lines 23-29). Moreover, the second phase also re-matches those Phase-1 results whose similarities are not sufficiently qualified by finding another high-similarity frame in the frame index range bounded by two qualified matching frames found in Step 2.1 (Lines 30-48). The Phase-2 results are then stored in ${}^{[F|L|R]}\mathbf{P2R}$. The last phase, called the integration phase, (Lines 49-56) is to integrate the Phase-2 results of the three ALPS models by selecting frames of the highest similarity among the Phase-2 results ${}^{[F|L|R]}\mathbf{P2R}$. The result $\hat{Z}^* = \{\beta_i | i = 1, \ldots, K\}$ (Line 56), containing the best frame index β_i in convalescent's clip that match to all α_i, is the solution found for Eq.(8) of the PVM problem.

V. Case Study

A. Environmental Settings and Performance Metrics

We developed the proposed PVM method for the case study using Python and PyTorch. The pretrained models of Alphapose [6] and DST [7] are adopted to implement $\mathfrak{F}_{\text{PHP}}$ and $\mathfrak{F}_{\text{3DPHP}}$ in the developed prototype. The parameters θ_{sim} and ϕ used in Algorithm 1 are set to 0.75 and 15%, respectively. The performance measurement adopts the mean absolute error (MAE), which measures the offset between the ground-truth frames and those found by a method. Let y_i, $i = 1, \ldots, K$ be the frame index of ground truths matching to anchor frames α_i.

[5]The symbol ${}^{[F|L|R]}\mathbf{P1R}$ is a short-cut representation of three data structures: ${}^{F}\mathbf{P1R}$, ${}^{L}\mathbf{P1R}$, and ${}^{R}\mathbf{P1R}$.

Algorithm 1: Three-Phase ALPS Matching Algorithm

/* Initialization: Construct ${}^{F}\mathbf{Z}$, ${}^{L}\mathbf{Z}$, ${}^{R}\mathbf{Z}$, the element difference matrix of ALPS similarity between mentor and convalescent for solving Eq.(8) in three perspectives with considering the temporal decay factor. */

1 $\rho_i = \frac{\alpha_i - \alpha_{i-1}}{M} \cdot N$, for $i = 1, \ldots, K$; // estimated i-th movement interval in convalescent's clip.

2 **foreach** *((i, j), where $i \in [1, K]$ and $j \in [1, N]$)* **do**

3 $\tau_{i,j} = |\sum_{k=1}^{i} \rho_k - j|$; // The distance between client frame j and the referential index estimated from anchor distribution.

/* temporal matching decay parameter $tmd(i, j)$: for frame j exceeding the estimated region of i-th movement (measured by $\tau_{i,j}$), certain punishment is added to $\mathbf{Z}_{i,j}$. */

4 $tmd(i, j) = 1 + [\![\tau_{i,j} > R_i]\!] \cdot \left(\frac{\tau_{i,j} - R_i}{R_i}\right)^2$, where $R_i = \rho_i \cdot \frac{\sigma}{\mu}$, $\mu = \frac{1}{K}\sum_{i=1}^{K} \rho_i$, $\sigma = \left(\frac{1}{K}\sum_{i=1}^{K}(\rho_i - \mu)^2\right)^{1/2}$;

5 **foreach** *($Type \in \{F, L, R\}$)* **do**

6 ${}^{Type}\mathbf{Z}_{i,j} = \left(1 - \cos({}^{Type}_{m}\mathbf{A}^{\alpha_i}, {}^{Type}_{c}\mathbf{A}^{j})\right) \times tmd(i, j)$;

7 **end foreach**

8 **end foreach**

9 Phase 1: (Rough-Matching Phase) Obtain one-to-multiple matching sets for three $\mathbf{Z}$ matrices.

// Step 1.1. Construct three $\mathbf{Z}$'s based on DTW [14] with boundary conditions: (1) $\mathbf{Z}_{0,0} = 0$ and (2) $\mathbf{Z}_{i,j} = \infty$ if either $i = 0$ or $j = 0$.

10 **foreach** *($Type \in \{F, L, R\}$)* **do**

11 **foreach** *((i, j), where $i \in [1, K]$ and $j \in [1, N]$)* **do**

12 ${}^{Type}\mathbf{Z}_{i,j} = {}^{Type}\mathbf{Z}_{i,j} + \min({}^{Type}\mathbf{Z}_{i-1,j}, {}^{Type}\mathbf{Z}_{i,j-1}, {}^{Type}\mathbf{Z}_{i-1,j-1})$;

13 **end foreach**

14 **end foreach**

// Step 1.2. Find Phase-1 result $P1R$ by backtracking minimum cost[14].

15 $i = K + 1$; $j = N + 1$; ${}^{F}\mathbf{P1R} = {}^{L}\mathbf{P1R} = {}^{R}\mathbf{P1R} = \emptyset$;

16 **foreach** *($Type \in \{F, L, R\}$)* **do**

17 **while** *($i > 0$ or $j > 0$)* **do**

18 $(i, j) = \begin{cases} (i-1, j-1), & \text{if } {}^{Type}\mathbf{Z}_{i-1,j-1} < {}^{Type}\mathbf{Z}_{i,j-1}, \\ (i, j-1), & \text{if } {}^{Type}\mathbf{Z}_{i-1,j-1} \geq {}^{Type}\mathbf{Z}_{i,j-1}. \end{cases}$

19 ${}^{Type}\mathbf{P1R} = {}^{Type}\mathbf{P1R} \cup \{(i, j)\}$;

20 **end while**

21 **end foreach**

22 Phase 2: (Fine-Matching Phase) Obtain one-to-one and qualified matching pair sets for Phase-1 results.

// Step 2.1. Ensuring only one frame matches to α_i, as Step 1.2 could incur many-to-one matches. (one-to-one property)

23 **foreach** *($Type \in \{F, L, R\}$)* **do**

24 **for** *($i = 1$ to K)* **do**

25 ${}^{Type}MatchSet[i] = \{j \mid (i, j) \in {}^{Type}\mathbf{P1R}.\}$; // Collect client's frame j that matches to anchor i in Step 1.2.

26 $\beta_i = \underset{j \in {}^{Type}MatchSet[i]}{\arg\max} \cos({}^{F}_{m}\mathbf{A}^{\alpha_i}, {}^{F}_{c}\mathbf{A}^{j})$;

27 ${}^{Type}FirstPassMatching[i] = \left(\beta_i, \cos({}^{F}_{m}\mathbf{A}^{\alpha_i}, {}^{F}_{c}\mathbf{A}^{\beta_i})\right)$;

28 **end for**

29 **end foreach**

// Step 2.2. For matches not satisfying the similarity threshold, find another high-similarity frame in the proper range. (qualified property)

30 Let ${}^{F}\mathbf{P2R} = {}^{L}\mathbf{P2R} = {}^{R}\mathbf{P2R} = \emptyset$ and θ_{sim} be the matching quality threshold;

31 **foreach** *($Type \in \{F, L, R\}$)* **do**

32 **foreach** *($(\beta_i, sim_{\beta_i}) \in {}^{Type}FirstPassMatching$), with index i)* **do**

33 **if** *($sim_{\beta_i} < \theta_{sim}$)* **then**

// β_i in the first-pass matching is underqualified so α_i will be re-matched in the following ''else'' part.

34 $LowSimAnchor$.append(α_i);

35 continue; // Directly go to next for-loop iteration of Line 32.

36 **else** // Re-match Phase-1 results whose similarities $< \theta_{sim}$.

37 **for** *($\alpha' \in LowSimAnchor$)* **do**

/* Determine start and end index pair $(r^{\vdash}, r^{\dashv})$ of convalescent's frame range for matching α'. */

38 $(r^{\vdash}, r^{\dashv}) = (\text{Tail}(P2R) + \sigma, \beta_i - \sigma)$;

39 $T = \{k \big| k \in [r^{\vdash}, r^{\dashv}] \text{ and } \cos({}^{F}_{m}\mathbf{A}^{\alpha'}, {}^{F}_{c}\mathbf{A}^{k}) \geq \text{PercentRank}(\phi, \{cos({}^{F}_{m}\mathbf{A}^{\alpha'}, {}^{F}_{c}\mathbf{A}^{i}), i = r^{\vdash}, \ldots, r^{\dashv}\})\}$; // PercentRank$(\phi, X)$: return $x_i \in X$ ranked at top ϕ percent of element values in X.

40 $\theta_\phi = \frac{1}{|T|}\sum_{k \in T} cos({}^{F}_{m}\mathbf{A}^{\alpha'}, {}^{F}_{c}\mathbf{A}^{k})$;

/* Obtain frame β' matching to α' by finding the first frame whose similarity $\geq \theta_\phi$ in range $(r^{\vdash}, r^{\dashv})$. */

41 $\beta' = \underset{k \in [r^{\vdash}, r^{\dashv}]}{\arg\min} \cos({}^{F}_{m}\mathbf{A}^{\alpha'}, {}^{F}_{c}\mathbf{A}^{k}) \geq \theta_\phi$:

42 ${}^{Type}\mathbf{P2R}$.append($\beta', \cos({}^{F}_{m}\mathbf{A}^{\alpha'}, {}^{F}_{c}\mathbf{A}^{\beta'})$);

43 **end for**

44 remove all elements in $LowSimAnchor$;

45 **end if**

46 ${}^{Type}\mathbf{P2R}$.append(β_i, sim_{β_i});

47 **end foreach**

48 **end foreach**

49 Phase 3: (Integration Phase) Obtain the final matching result from ${}^{F}\mathbf{P2R}$, ${}^{L}\mathbf{P2R}$, and ${}^{R}\mathbf{P2R}$.

50 $\hat{Z}^* = \emptyset$; // repository of the final matching results.

51 **for** *(i=1 to K)* **do**

/* Find out $TMax$, the Type (F, L, or R) with maximal similarity, among three Phase-2 results for the i-th movement. */

52 $TMax = \text{array}(\underbrace{{}^{F}\mathbf{P2R}[i]}_{\text{case F}}, \underbrace{{}^{L}\mathbf{P2R}[i]}_{\text{case L}}, \underbrace{{}^{R}\mathbf{P2R}[i]}_{\text{case R}})$.argmaxsim();

/* Determine the final frame index β_i by finding the maximal similarity from the Full, Left, and Right matching scores. */

53 $\beta_i = \begin{cases} {}^{F}\mathbf{P2R}[i].\text{getbeta}(), & \text{if } TMax\text{==F}, \\ {}^{L}\mathbf{P2R}[i].\text{getbeta}(), & \text{if } TMax\text{==L}, \\ {}^{R}\mathbf{P2R}[i].\text{getbeta}(), & \text{if } TMax\text{==R}. \end{cases}$

54 $\hat{Z}^* = \hat{Z}^* \cup \{\beta_i\}$; // Preserve the frame index matching to α_i.

55 **end for**

56 **return** $\hat{Z}^*$; // Solution for Eq.(8)

Assuming that $\hat{Z}^* = \{\beta_i\}$ are the frame index by a matching method, the MAE is computed as:

$$\text{MAE} = \left| \frac{\sum_{i=1}^{K} y_i - \beta_i}{K} \right| \tag{22}$$

To validate the effectiveness of the proposed method, we collected real-world physiotherapy video data from our collaborating hospital in Tainan, Taiwan. All video clips are recorded at the industrial frame rate of 30 frames/second. The mentor, who has a national nurse license in Taiwan, performs 11 widely used physiotherapy movements (i.e., $K = 11$) in a clip of 19.6 seconds (588 frames). The convalescent performs the mentor's physiotherapy movements in a clip of 20 seconds (601 frames). While recording video clips, cameras are placed in front of the mentor and in the convalescent's three different angles (front, left, right) to test matching methods.

B. Qualitative Study

We apply the proposed method to real-world test scenarios, which are also used in the following quantitative studies. Fig. 5 shows a screenshot of our demo video, where the complete comparisons can be found at our GitHub: https://github.com/NCKU-CIoTlab/TALMA-on-ALPS/. Three camera-angle pairs of the mentor and the convalescent, including front-front, front-left, and front-right, are used to test matching performance in different camera-angle scenarios. Notably, the convalescent did not perform the eleventh movement, and our method also identified this case by giving a low similarity score.

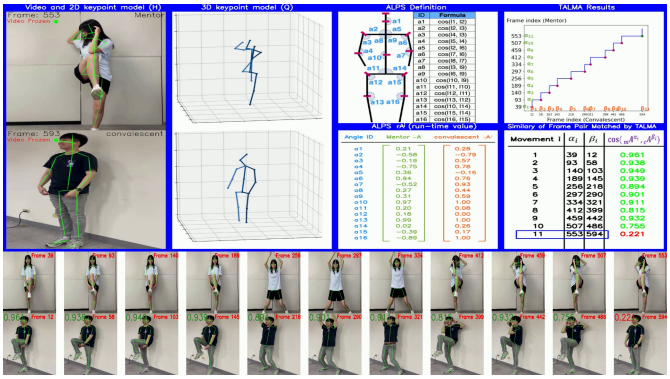


Fig. 5: A screenshot in the demo video illustrating the proposed PVM method.

C. Precision Study

Table I shows the MAE of related methods in different camera-angle settings, which are represented in the mentor-convalescent pair with F, L, R meaning the camera position is Front, Left, and Right, respectively, to the mentor or the convalescent. For fair comparisons, all seven methods use the same algorithm structure. This experiment can also be treated as a kind of ablation study. Among the seven methods, our proposed method in the last row has the least MAE (1.6 in overall) and the least variance (1.2-2.1 in all camera-angle settings), which shows the high precision and robustness of our mechanisms. Note that MAE=2.1 means that the time difference between our method and the ground truths in the convalescent clip is less than $2.1/30 \leq 0.07$ seconds, indicating that our results and the ground truths are almost indistinguishable from human experts.

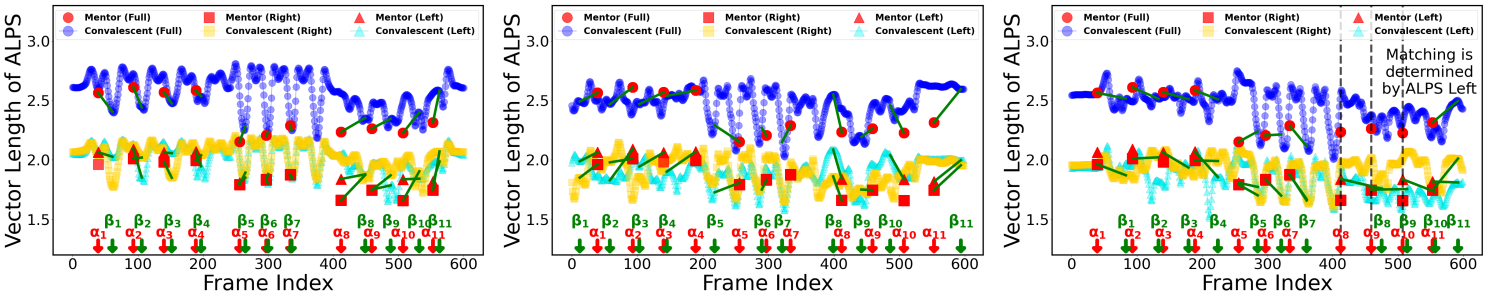
More detailed analysis is described below. The last two rows of Table I show the advantage of adopting three ALPS models in Algorithm 1, since the ^{F}ALPS+TALMA produces higher MAE in the F-R case. We will further analyze this by its matching behavior in the following experiment. Besides, the 2D keypoint-based methods (rows 1 and 4-5) incurring high MAE show that less-quality matches are produced, even if they work with the TALMA. The MAE results of the 2D keypoint-based methods (rows 4-5) are also more unstable (1.9 in F-F, while ≥ 8.1 and ≥ 10.8 in F-L and F-R, respectively) than those of last two rows, which reveals the stability of ALPS. Rows 1-3 give the matching performance using DTW [14], showing that traditional matching algorithms without considering physiotherapy semantics perform worse than those with the TALMA (rows 4-7), even if they work with the ALPS (referring to rows 2-3). The results show that TALMA is critical to achieve high precision. In addition, we have studied the impact of parameters θ_{sim} and ϕ used in Step 2.2 of Algorithm 1 to find proper matches, and the results are presented in the technical report [15] due to the length limit.

TABLE I: The mean absolute error (MAE) comparison across different methods and camera positions. In the methods, 2D keypoints are obtained by Alphapose [6] and DTW is the re-implementation from [14] with our structure definitions.

Method \ Camera Positions	F-F	F-L	F-R	Overall
^{F}2D-Keypoints+DTW	145.5	105.2	134.9	128.5
^{F}ALPS+DTW	82.3	112.3	31.4	75.3
$^{[F\|L\|R]}$ALPS+DTW	88.9	175.6	87.4	117.3
^{F}2D-Keypoints+TALMA	1.9	8.7	10.9	7.1
$^{[F|L|R]}$2D-Keypoints+TALMA	1.9	8.1	10.8	6.9
^{F}ALPS+TALMA	**1.2**	**2.1**	2.2	1.8
$^{[F\|L\|R]}$ALPS+TALMA	**1.2**	**2.1**	**1.7**	**1.6**

D. Interpretation of TALMA Behavior

Fig. 6 shows the matching relationship between the mentor's anchor frames and the convalescent clip to demonstrate the behavior of the TALMA. The horizontal axis is the frame index. The vertical axis is the length of an ALPS vector, i.e., mapping the vector $\mathbf{A}_{16\times 1}$ to $\|\mathbf{A}\|_2 = (\mathbf{A}^\top \mathbf{A})^{\frac{1}{2}} \in \mathbb{R}$, to visualize the matching relationship in a two-dimensional plot. The 11 anchor frames $\alpha_1, \ldots \alpha_{11}$ and the PVM results $\hat{Z}^* = \{\beta_1, \ldots, \beta_{11}\}$ obtained by Algorithm 1 are marked in red and green, respectively, in three subfigures. The ALPS model pairs for the matching results (α_i, β_i), $i = 1, \ldots, 11$, are connected by green lines to indicate the TALMA decisions. Notice that β_i could be found out from multiple ALPS models simultaneously, in case their similarity values to the associated anchor frame are equal. For example, in Fig. 6(a), similarity scores of α_1 and β_1 on $^{[F|L]}$ALPS are equal in Line 52 of Algorithm 1, so



(a) Front-Front (b) Front-Left (c) Front-Right

Fig. 6: Visualizing TALMA matching behavior for physiotherapy applications, where the camera angles of the mentor and the convalescent are front-front, front-left, and front-right, respectively. The situations marked in vertical dashed lines in Fig. 6(c) illustrates the TALMA of three ALPS models finds out better matches than that of only ^{F}ALPS.

green lines between α_1 and β_1 on Mentor (Full) and Mentor (Left) are drawn. This figure pictures Step 3 of Algorithm 1.

From the results of Fig. 6(a)-Fig. 6(c), 30 out of 33 anchor frames in the test clips are successfully matched by using merely the ^{F}ALPS, which give the reason of row 5 (^{F}ALPS+TALMA) in Table I reaching similar performance of row 6. The main difference between these two methods occurs in matching anchor frames α_8-α_{10} in Fig. 6(c) (marked with dashed vertical lines), where ^{L}ALPS provides the higher similarity score than ^{F}ALPS. This means that precisely identifying these convalescent movements needs to pay more attention to the left-side body features due to the camera-angle effect. Our method adopts three ALPS models and thus has robust video matching capability to cope with various positions of camera placement.

VI. Conclusions and Future Work

"Hospital at home" enlightens revolutions of medical service automation. This paper revisited the physiotherapy service so that convalescents at home are able to record their physiotherapy clips with an arbitrarily placed mobile device for online verification. In our paper, we proposed a physiotherapy video matching method supporting arbitrary camera placement, so that physiotherapy movements in video clips of free camera shoot angles can be accurately matched. The key contributions are summarized as follows: (1) Sec. II formulates the PVM problem in an optimization form, which is then solved by our designed pipeline in Sec. III with existing AI models (Alphapose and DST) for rapid development, (2) the ALPS and associated CAFE transformation are invented to model human movements for alleviating the camera-angle effect, and (3) the TALMA is developed to effectively solve the physiotherapy video matching problem. Our real-world experiments validate that the proposed method indeed performs superior in precision and practicality. The explicable results also verify the necessity of our mechanisms, including ALPS and TALMA. This study provides a helpful reference for industrial practitioners to develop pragmatic remote hospital services with solid theoretical foundations for increasing their business competitiveness.

Our future work will focus on reducing the processing time of AI models to accommodate real-time applications. In addition, studying the relationship between parameters θ_{sim} and ϕ is a practical direction to increase the automation degree of the TALMA for applying our method to other industries.

References

[1] J. S. Bolwell, "A Review of Healthcare Challenges in the UK and the US: Medical Errors, Aging, Private Healthcare and Governance," *Health Sciences Review*, p. 100211, 2025.

[2] W. Kim, "Exploring the Structural Reform of Youth Policies to Promote Fertility," International Center for Public Policy, Andrew Young School of Policy Studies, Georgia State University, Working Paper 2501, 2025. [Online]. Available: https://ideas.repec.org/p/ays/ispwps/paper2501.html

[3] E. Abbott, A. Campbell, and S. Tidman, "Reliability of Single-Plane Movement Observations in Physiotherapy Assessments," *Rehabilitation and Posture Analysis*, vol. 29, no. 4, pp. 301–315, 2023.

[4] R. Ruivo, P. Pezarat-Correia, and A. Carita, "Static Posture Assessment Using Software Analysis," *Posture Science*, vol. 15, no. 4, pp. 202–215, 2023.

[5] F. Mortazavi and A. Nadian-Ghomsheh, "Stability of Kinect for Range of Motion Analysis in Static Exercises," *Physiotherapy Technology*, vol. 14, no. 5, pp. 345–360, 2023.

[6] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body Regional Multi-person Pose Estimation and Tracking in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7157–7173, 2022.

[7] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges, "A Spatio-temporal Transformer for 3D Human Motion Prediction," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 565–574.

[8] A. Z. Skouras, A. K. Kanellopoulos, S. Stasi, A. Triantafyllou, P. Koulouvaris, G. Papagiannis, and G. Papathanasiou, "Clinical Significance of the Static and Dynamic Q-angle," *Cureus*, vol. 14, no. 5, 2022.

[9] P. Kejonen and K. Kauranen, "Reliability and Validity of Motion Analysis for Standing Balance Measurements," *Balance and Posture Analysis*, vol. 22, no. 3, pp. 140–155, 2023.

[10] N. Stergiou and R. Harbourne, "Optimal Movement Variability in Neurologic Physiotherapy," *Neurologic Rehabilitation*, vol. 11, no. 2, pp. 78–92, 2023.

[11] P. Levinger and W. Gilleard, "Postural Assessment and Foot Motion Analysis," *Postural Analysis*, vol. 20, no. 6, pp. 350–365, 2023.

[12] E. Ronn, "NP-complete stable matching problems," *Journal of Algorithms*, vol. 11, no. 2, pp. 285–304, 1990.

[13] C.-C. Chen, M.-H. Hung, B. Suryajaya, Y.-C. Lin, H.-C. Yang, H.-C. Huang, and F.-T. Cheng, "A Novel Efficient Big Data Processing Scheme for Feature Extraction in Electrical Discharge Machining," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 910–917, 2019.

[14] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[15] J.-W. Lin, P.-W. Chen, Y.-P. Huang, M.-H. Hung, M.-H. Kao, J. Ji, L.-Y. Jiang, Y.-S. Chou, and C.-C. Chen, "A Physiotherapy Video Matching Method Supporting Arbitrary Camera Placement via Angle-of-Limb-based Posture Structures," *Technical Report, Available at https://github.com/NCKU-CIoTlab/TALMA-on-ALPS/*, 2025.

APPENDIX

A. COMPUTATIONAL EXPLANATIONS TO TALMA (ALGORITHM 1)

The purpose of this section is to help industrial practitioners to facilitate kernel development of remote hospital services with knowing the computational logic of the TALMA. The designs of the TALMA include leveraging existing matching methods, handling intermediate results of three ALPS perspectives, and enhancing match quality, which are sophisticatedly arranged in Algorithm 1. Hence, we extend the design discussion in Sec. IV-C and give detailed computational explanations to TALMA in this section.

In order to have a rough matching by the DTW method, Phase 1 constructs ${}^{F}\mathbf{Z}_{K\times N}, {}^{L}\mathbf{Z}_{K\times N}, {}^{R}\mathbf{Z}_{K\times N}$ as the element difference matrix of $\{{}^{F}_{m}\mathbf{A}^{\alpha_i}\}_{i=1}^{K}$ and $\{{}^{F}_{c}\mathbf{A}^{j}\}_{j=1}^{N}$, $\{{}^{L}_{m}\mathbf{A}^{\alpha_i}\}_{i=1}^{K}$ and $\{{}^{L}_{c}\mathbf{A}^{j}\}_{j=1}^{N}$, $\{{}^{R}_{m}\mathbf{A}^{\alpha_i}\}_{i=1}^{K}$ and $\{{}^{R}_{c}\mathbf{A}^{j}\}_{j=1}^{N}$, respectively (in Lines 2-8), for solving Eq.(8) with three ALPS models considering the temporal decay factor. The notation ${}^{Type}\mathbf{Z}$, $Type \in \{F, L, R\}$, is used for shortly representing the three matrices via the variable $Type$ throughout this algorithm. Element ${}^{Type}\mathbf{Z}_{i,j}$ computed by $\big(1-\cos({}^{Type}_{m}\mathbf{A}^{\alpha_i}, {}^{Type}_{c}\mathbf{A}^{j})\big) \times tmd(i,j)$ in Line 6 stands for weighted similarity of mentor frame i and convalescent frame j, where the similarity term $\big(1-\cos({}^{Type}_{m}\mathbf{A}^{\alpha_i}, {}^{Type}_{c}\mathbf{A}^{j})\big)$ is set according to Eq. (8) for solving the PVM problem and the penalty term $(tmd(i,j))$ is added to satisfying the physiotherapy scenarios as a convalescent performing movement by mimicking the mentor's online movements shall be in a period similar to the mentor's clip (temporal property). Computing $tmd(i,j)$ in Line 4 with estimated mean μ and variance σ of the i-th movement interval in the convalescent clip is designed to express the physiotherapy semantics: a distanced frame considering for matching needs to pay the penalty, which is proportional to the distance to the referential frame whose position is the same position to the anchor frame in the mentor clip. After connecting the element distance matrix Z to our concerned PVM problem, the rest steps in Phase 1 (Lines 10-21) follow the DTW algorithm [14] to yield the rough-matching results ${}^{[F|L|R]}\mathbf{P1R}$.

Phase 2.1 is designed to provide the one-to-one property and the matching quality property for the Phase-1 matching results of three ALPS models, where the many-to-one property of the Phase-1 results inherits from the DTW algorithm. To achieve the one-to-one property, The redundant matches to each of the K anchor frames in the Phase-1 results of each ALPS model are trimmed by evaluating the full-body similarity (Lines 23-29). Notice that the full-body similarity considers all feature perspectives of a human posture, for not emphasizing left-side or right-side features without knowing any more rehabilitation semantics. The preserved convalescent frame is the one that has the highest full-body similarity to the corresponding anchor frame. Thus, finding maximal full-body similarity between the anchor frame α_i and a convalescent frame, i.e., $\underset{j\in {}^{Type}MatchSet[i]}{\arg\max} \cos({}^{F}_{m}\mathbf{A}^{\alpha_i}, {}^{F}_{c}\mathbf{A}^{j})$ in Line 26, is the key computational step to trim redundant frames matching to α_i.

Phase 2.2 aims to provide better matching quality for the results of Phase 2.1 (Lines 30-48). To enhance the matching quality, we use the expert-automation collaborating principle: human experts provide their desired matching standard via the parameter θ_{sim}, and algorithmic procedure handles those frames that cannot satisfy experts' specified matching standard in an automated manner. Our Phase 2.2 is designed in a greedy manner with the assistance of the variable $LowSimAnchor$, maintaining the under-qualified anchor frames whose similarity is less than θ_{sim}. The procedure sequentially verifies matching candidates in the Phase-2.1 results of the three ALPS models (Lines 30-31). If the candidate is below the similarity threshold θ_{sim}, then the corresponding anchor frame (i.e., under-qualified anchor frame) is appended into $LowSimAnchor$ (Line 33-34), meaning that this anchor frame needs to be re-matched later. In order to process the under-qualified anchor frames in a batch, the re-matching procedure of the greedy design will be launched until the next qualified anchor frame is found. Specifically, the iteration for an under-qualified match $(\beta_i, sim_{\beta_i}$ in the loop of Line 32 shall be terminated and the next iteration of $(\beta_i, sim_{\beta_i}$ of the loop shall then directly continue. Thus, the "continue;" statement (Line 35) is used to implement this purpose. In case the iteration of $(\beta_i, sim_{\beta_i}$ in the loop of Line 32 satisfies the threshold θ_{sim} (Line 36), then this candidate match will be preserved in the Phase 2 result and the under-qualified anchor frames in $LowSimAnchor$ start the re-matching procedure (Lines 37-46). As a convalescent could perform physiotherapy exercises in any dissimilar movements, the re-matching threshold needs to be automatically computed considering the movement status in the current convalescent clip. Hence, the key of the re-matching procedure is to determine a proper re-matching quality threshold for re-evaluating the convalescent frames to match the anchor frames in $LowSimAnchor$. Our idea is to sample the top ϕ percent of the convalescent frames for filtering noisy convalescent frames to re-match the current under-qualified anchor frames (Line 39), where the search range $(r^{\vdash}, r^{\dashv})$ (Line 38) is thus needed to calculate the top ϕ percent of the samples from the convalescent frames. The proper threshold θ_ϕ based on selected top-ϕ-percent frames in the search range $(r^{\vdash}, r^{\dashv})$ on is computed in Line 40 by calculating the mean similarity score of the selected frame set T. Once θ_ϕ is obtained, the re-matching is performed with the threshold θ_ϕ in Line 41, where the matching of under-qualified anchor frames is now refined by re-finding the better convalescent frames in the range $(r^{\vdash}, r^{\dashv})$ than those in Phase 2.1. One may further to determine the θ_ϕ by considering the similarity of anchor frames in $LowSimAnchor$, instead of the overall evaluation like our current version in Line 39. The following sections will show that the current design is sufficiently effective and we leave this extension to industrial participants if their needs require such modification.

The Phase 3 is to integrate the Phase-2 results of the three ALPS models by selecting frames of the highest similarity among the Phase-2 results ${}^{[F|L|R]}\mathbf{P2R}$ corresponding to the three ALPS models (Lines 50-56). Note that the member function "argmaxsim()" in Line 52 will return the index of the $array(.)$, which is recorded by the $TMax$. The matching result β_i is then retrieved from one of i-th element in Phase-2 results (i.e., ${}^{F}\mathbf{P2R}[I], {}^{L}\mathbf{P2R}[i], {}^{R}\mathbf{P2R}[i]$) by using the value

of $TMax$ in Line 53. In this way, the matching frames are ultimately found and maintained in $\hat{Z}^*$ in Line 54.

B. Impact of Matching Quality Threshold θ_{sim}

Discussing the impacts of parameters θ_{sim} and ϕ is highly related to the cases happening in Algorithm 1. Thus, we leave the two studies in the appendix.

We study the experimental results of the parameter θ_{sim} used in Step 2.2 of Algorithm 1 and the results are shown in Table II. In the table, #LSA stands for the number of the re-matching process triggered for α_i in $LowSimAnchor$ (referring to Line 33 in the algorithm), where such the α_i happens as the associated similarity sim_{β_i} is less than θ_{sim}. From the results, we can see that #LSA increases as the increasing θ_{sim} for all methods, as a great θ_{sim} value decreases the number of Phase-1 matching anchor frames whose similarity exceeding θ_{sim} in Line 33 of Step 2.2, which then makes #LSA increases. Note that setting θ_{sim} to a proper value (e.g., 0.75 in the results) reaches optimal performance. On the one hand, a great θ_{sim} value selects only a few amounts of high similarity matched frame pair to the Phase-2 result. This yields a great number of unmatched anchor frames in #LSA, which makes less quality of re-matching results in the "else" block of Lines 36-45. On the other hand, a small θ_{sim} value selects the great amount of low similarity matched frame pair to Phase-2 result, which increases the MAE in the matching results. Thus, a proper great θ_{sim} value (i.e., 0.75 in our experiments) leaving a sufficient amount of unmatched anchor frames to be re-matched in customized matching threshold in Lines 36-45 reaches optimal performance.

The fact of row 3-4 performing better than rows 1-2 comes from that matching on ALPS is easier to satisfy θ_{sim} than 2D keypoint structures, making the former has more number of high similarity matched frame pairs than the later. This supports that the ALPS is a better representation than 2D-keypoint-based structures to work with TALMA. In addition, the row 4 performs better than the row 3 as three ALPS instances are used to find sufficiently high similarity matching results over θ_{sim}, which thus has more chances to find better matches.

C. Impact of re-matching control parameter ϕ

We study the impact of the parameter ϕ (portion of high similar frames for computing the re-matching threshold) used in Step 2.2 of Algorithm 1, and the results are shown in Table III. As our expectations, MAE values increase as increasing ϕ in all methods, since the large ϕ would select a great number of frame candidates in the search range of convalescent video to compute the re-matching quality threshold θ_ϕ , which decreases θ_ϕ, and thus yields large MAE. In the results, it is observed that ALPS-based methods (rows 3-4) perform better than 2D-keypoint-based methods (rows 1-2), which is also a support to verify the necessity of ALPS. Whereas in some situations, the MAE values at small ϕ (such as $\phi < 5\%$ in this experiment) are higher than those in many other large ϕ cases. This can be explained via Fig. 6(c). While multiple unmatched anchor frames[6] are appended in the $LowSimAnchor$, the small ϕ may make the re-matching quality threshold θ_ϕ be computed by considering many convalescent frames that have high similarity but shall be matched to other unmatched anchor frames in the view of ground truths. Thus, the small ϕ generates an inappropriate θ_ϕ, which then produces low quality matches and incurs large MAE, such as $\phi \leq 4\%$ in the 2D-keypoint-based methods (rows 1-2) and $\phi \leq 5\%$ in the ALPS-based methods (rows 3-4) in the table.

The studies of the two parameters θ_{sim} and ϕ are to show that the appropriate values are adopted in our experiments. Systematically finding out their values is worth deeply studying, but it is a bit out of this paper's scope. So, we leave this issue as one of the future research issues.

[6]This situation happens particularly when these unmatched anchor frames are similar.

TABLE II: Performance comparison of different methods in #LSA and the mean absolute error (MAE) across varying the matching threshold θ_{sim}. Here, #LSA stands for the number of the re-matching process triggered for α_i in $LowSimAnchor$.

Method \ θ_{sim}		0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
^{F}2D-Keypoints + TALMA	#LSA	4	5	6	10	11	12	13	17	21	23
	MAE	11.80	11.87	11.57	7.66	7.43	**7.16**	7.23	8.70	13.90	14.26
$^{[F|L|R]}$2D-Keypoints+TALMA	#LSA	10	15	17	28	31	33	40	52	64	70
	MAE	9.19	9.39	9.39	6.63	6.76	**6.93**	7.03	8.46	13.90	14.26
^{F}ALPS + TALMA	#LSA	1	3	3	3	4	5	6	7	9	22
	MAE	9.23	4.50	4.50	4.50	2.93	**1.83**	1.86	2.00	2.03	9.83
$^{[F\|L\|R]}$ALPS + TALMA	#LSA	6	10	10	10	14	18	20	23	35	66
	MAE	3.03	3.03	3.03	3.03	2.76	**1.66**	1.70	1.83	2.03	9.83

TABLE III: Performance comparison of different methods in the mean absolute error (MAE) across varying the percentage threshold ϕ under the matching quality threshold θ_{sim} of 0.75.

Method \ ϕ	3%	4%	5%	15%	25%	35%	45%	55%	65%	75%	85%	95%
^{F}2D-Keypoints + TALMA	11.60	20.27	**6.43**	7.16	9.00	9.46	13.30	16.03	16.73	17.86	25.09	29.20
$^{[F|L|R]}$2D-Keypoints+TALMA	11.56	20.23	**6.39**	6.93	8.53	8.86	12.50	15.06	15.76	16.86	18.09	19.06
^{F}ALPS + TALMA	6.66	6.53	6.43	**1.83**	2.40	2.80	3.03	3.16	3.30	3.46	3.86	8.29
$^{[F\|L\|R]}$ALPS + TALMA	6.56	6.43	6.36	**1.66**	1.96	2.23	2.33	2.40	2.46	2.50	2.50	2.53