Personalized Markerless Upper-Body Tracking with a Depth Camera and Wrist-Worn Inertial Measurement Units*

Prayook Jatesiktat[1], Dollaporn Anopas[1], and Wei Tech Ang[1]

***Abstract*— A markerless motion capture technique is proposed based on a fusion between a depth camera (Kinect V2) and a pair of wrist-worn inertial measurement units (IMU). The method creates a personalized articulated human mesh model from one depth image frame and uses that model to improve the accuracy of the upper-body joint tracking. The IMUs are useful as an additional clue for the arm tracking, especially during an occlusion. An evaluation of the method against a marker-based system as a gold standard using data from 6 subjects is done. The result shows over 20% reduction in upper-limb joint position errors when compared to Kinect's skeleton tracking. All the collected data are calibrated, synchronized, and made publicly available for research purposes.**

I. Introduction

Human pose estimation from a single depth camera has become a popular research topic due to the emergence of low-cost depth sensors such as Kinect (Microsoft, USA) in 2010. Current state-of-the-art techniques can be divided into discriminative and generative methods. Generative methods search in a pose space to fit a template to the observation. The model could be in various forms such as a mesh [1], a sphere set [2], or articulated Gaussian kernels [3]. Those purely generative methods always rely on the result from the previous frames. Therefore, it is hard to recover when the tracking is lost from a fast movement. This leads to the discriminative initialization to improve the robustness of the tracking. A common way to initialize the pose parameter is to turn the observed frame to a low-dimensional feature vector to allow a quick search in a discrete pose database [4], [5], [6] and then refine the result using a generative method.

However, two purely discriminative learning-based methods [7] have achieved good joint prediction accuracy using a per-pixel classification or a direct joint position regression. One of them was implemented into Kinect software development kit (SDK) and was evaluated multiple times in different scenarios. On Kinect V2, the average upper limb joint position error is in the range of 50-101 mm and they will get off-track at 1-5% of the time [8]. This can be interpreted to 1-5 frames every 3.3 seconds (at 30 fps) that a joint will jump out wrongly. Those errors should be addressed in some sensitive applications such as rehabilitative assessment.

This work aims to improve the skeleton tracking for upper-limb rehabilitative exercise assessment tasks. Our body-tracking solution is inspired by an emerging hand tracking technique from Microsoft [9] together with an availability of an anatomically-accurate human mesh model which can morph its shape to fit individual users [10]. Our method fits an articulated mesh model to the depth image sequence using an optimization method that updates both pose parameters and correspondence matching simultaneously [9]. In addition, 2 additional wrist-worn IMUs are integrated in the optimization to add a forearm occlusion recovery feature.

Using IMUs to improve a visual-based pose estimation was first introduced in the context of multi-view silhouettes from multiple RGB cameras [11]. However, with a single depth camera, IMU orientations have only been used for a table lookup in the discriminative step [6]. We apply them differently by using IMU information directly in the cost function of the generative step. Importantly, as we target rehabilitative patients as our main user group, requiring an IMU on their head or at the back of their trunk could be difficult in a home-based rehab setup. Therefore, a less obtrusive configuration of 2 wrist-worn IMUs together with a depth camera is purposed.

As we target at home-based rehab exercise assessment, some subtle details that are usually ignored in other pose tracking methods are addressed in this work. For example, the method should be able to detect the scapular elevation because it is a common compensatory movement that a stroke patient should be encouraged to avoid. The forearm pronation/supination should also be tracked as they can infer the state of recovery. It has never been tracked reliably with a depth-based visual motion capture system.

As model personalization is found to be significant for human pose estimation accuracy [2], an automatic shape personalization method is also implemented in our work with a similar concept to Helten *et al*.'s work [12] but based on a publicly-released statistical model [10].

II. Method

A. Frame Inputs & Expected Outputs

A Kinect V2 provides depth images at 30 Hz. For each depth frame, Kinect SDK is used to find the associated 3D point cloud, the body silhouette, and the 3D joint positions (Kinect skeleton). Three 9-axis IMUs are sampled at 80 Hz. Two of them are on both subject's wrists and one is mounted on the Kinect. This configuration can provide the orientations of the IMUs in the camera reference frame [13].

The final goal is to search for a pose θ that best fits to those observations. To evaluate the quality of fit, a triangulated surface will be rendered from an articulated human model called Skinned Multi-Person Linear model (SMPL) [10].

[1]Prayook Jatesiktat, Dollaporn Anopas, and Wei Tech Ang are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, 639798, Singapore. prayook001@e.ntu.edu.sg; anop0001@e.ntu.edu.sg; wtang@ntu.edu.sg

*This work was supported by the Second Rehabilitation Research Grant (RRG2/16001) from Rehabilitation Research Institute of Singapore (RRIS).

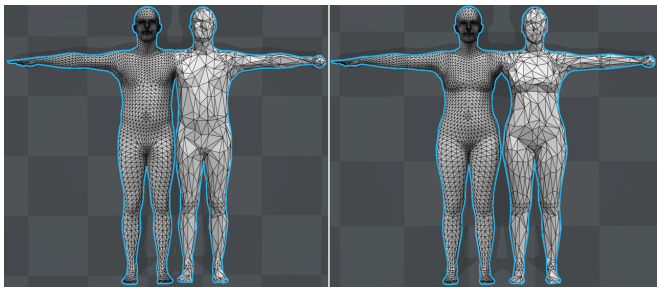


Fig. 1: Original high-resolution SMPL template meshes and the modified low-resolution SMPL template meshes. The male templates (left) and the female templates (right).

This model is driven by a pose (θ) and a personalized shape (β). Therefore, the problem can be divided into 2 phases. The first step is the model personalization which searches for both pose θ and shape β parameters from only one depth frame per subject (Section III). The personalized shape β from this process will then be used as a static value in the second step where we can only search for the pose θ (Section II-F).

B. Human Mesh Model

Unlike other statistical human models such as SCAPE [14] or Delta [1], the SMPL model renders its skin mesh in a closed-form calculation without any iterative optimization. This characteristic allows any points on the surface mesh to be differentiable with respect to shape and pose parameters.

The original SMPL model contains 6,890 vertices. Qslim [15] is used to reduce the resolution via vertex pair contraction. The software is set to prevent the contraction around the shoulders, the elbows, and the wrists to avoid artifacts from bending. Moreover, hands of the model are replaced by a round orb as this study does not intend to track hands. The final model contains 1,000 vertices (Fig.1).

Instead of using 3 degrees-of-freedom (DOF) in all the joints, we remove one DOF that represents the *roll* angle in every joint except the joints on the spine and the head. This constraint is applied to avoid the *candy wrapper artifact* when a joint twists due to the use of linear blend skinning. If needed, the *roll* information of the arm segments can be recovered later after the fitting process by using the IMU orientations. Our final model has 40 DOF for the upper-body model (from the hips and above) and 48 DOF for the full-body model including global translation and orientation. All DOF in a model are represented by $\theta \in \mathbb{R}^{N_\theta}$.

C. Parameterization of a Point on the Mesh

Any point on the mesh can be represented in the piecewise 2D parameter space

$$\Omega = \{(f,u,v) | f \in F; u \in [0,1]; v \in [0,1]; u+v \in [0,1]\} \tag{1}$$

where f is a triangular face on the mesh. (u,v) represents a barycentric coordinate (u,v,w) where $w = 1-u-v$. The set F contains all triangular faces on the articulated mesh.

Given an articulated mesh $\mathcal{P}(\theta) \in \mathbb{R}^{3\times M}$ rendered from the SMPL model with a specific shape vector β. The surface position function $S_p : \Omega \times \mathbb{R}^{N_\theta} \mapsto \mathbb{R}^3$ is defined by

$$S_p((f,u,v),\theta) = u\mathbf{p}_u + v\mathbf{p}_v + w\mathbf{p}_w \tag{2}$$

where $\mathbf{p}_u$, $\mathbf{p}_v$, and $\mathbf{p}_w$ are 3D position of 3 associated corners of triangle f on the articulated mesh $\mathcal{P}(\theta)$ and $w = 1-u-v$

For the surface normal function $S_\perp : \Omega \times \mathbb{R}^{N_\theta} \mapsto \mathbb{R}^3$, $S_\perp(\nu,\theta)$ is a unit normal vector from the mesh position ν on the model articulated by θ. Having the same normal vectors on one triangular face will zero the gradient of normal penalty term relative to u and v. This is avoided by calculating normal vector of each vertex from area-weighted sum of surrounding face normals [16]. Then, similar to the position function, we define the surface normal function as

$$S_\perp((f,u,v),\theta) = \text{II}(u\mathbf{n}_u + v\mathbf{n}_v + w\mathbf{n}_w) \tag{3}$$

where $\text{II}(\cdot)$ is a vector normalization. $\mathbf{n}_u$, $\mathbf{n}_v$, and $\mathbf{n}_w$ are associated normal vectors at the 3 corners of triangle f.

D. Energy Function

The energy function sums designed terms that give high value when the simulated model does not match with the observation or when a constraint is violated. It is defined as

$$E(\theta,\mathcal{U}) = E_{\text{data}}(\theta,\mathcal{U}) + \sum_{\tau\in\text{Terms}} \lambda_\tau E_\tau(\theta) \tag{4}$$

where Terms = {*bg*, *bgJoint*, *float*, *limit*, *imu*}.

1) data term: For each depth image, N_s pixels will be sampled from the body silhouette area to get 3D positions $\mathbf{p}_i$ and surface normals $\mathbf{n}_i$ (Section II-H). Each sampled point matches with a correspondent point ν_i on the piecewise 2D parameter space Ω. The point ν_i keeps moving to a better place in the process of the continuous update (Section II-I) and the discrete update (Section II-J).

Let $\mathcal{U} = \{\nu_i | \nu_i \in \Omega\}_{i=1}^{N_s}$ be a set of current correspondences. The *data* term penalizes the position and the normal differences between the sampled points from the depth image ($\mathbf{p}_i$, $\mathbf{n}_i$) and their correspondences (ν_i) on the model.

$$E_{\text{data}}(\theta,\mathcal{U}) = \frac{1}{N_s}\sum_{i=1}^{N_s}\left(\frac{\|S_p(\nu_i,\theta)-\mathbf{p}_i\|^2}{\sigma_p^2} + \frac{\|S_\perp(\nu_i,\theta)-\mathbf{n}_i\|^2}{\sigma_n^2}\right) \tag{5}$$

2) bg term: The *bg* term penalizes when some parts of the model are being projected outside the 2D silhouette.

$$E_{\text{bg}}(\theta) = \frac{1}{H}\sum_{h=1}^{H}\|\Pi_\triangleleft(S_p(\nu_h^{bg},\theta)) - \ni_{\text{2D}}(\Pi_\triangleleft(S_p(\nu_h^{bg},\theta)))\|^2 \tag{6}$$

$\mathcal{U}^{\text{bg}} = \{\nu_h^{\text{bg}}\}_{h=1}^{H} \subseteq \Omega$ is a set of selected points on the model which may cause the penalty. H points in this set are re-selected every iteration from the area of the model where the normal vector points perpendicular to the camera's optical axis. Those points will sparsely surround the model when projected to the 2D image and help to push the model closer to the silhouette.

$\Pi_{\triangleleft} : \mathbb{R}^3 \mapsto \mathbb{R}^2$ is a perspective projection from the 3D space to the 2D depth image space.

$\eth_{\text{2D}} : \mathbb{R}^2 \mapsto \mathbb{R}^2$ maps a projected 2D position on the depth image to a 2D position of the closest pixel on the silhouette. If the input is in the silhouette, $\eth_{\text{2D}}$ will act as an identical function. This function is pre-calculated on a distance transform using Quasi-Euclidean distance metric.

3) bgJoint term: This term penalizes when positions from 16 selected joints ($\mathcal{J}_{bg}$) are being projected outside the silhouette area.

$$E_{\text{bgJoint}}(\theta) = \frac{1}{|\mathcal{J}_{bg}|} \sum_{j \in \mathcal{J}_{bg}} \|\Pi_{\triangleleft}(\mathscr{J}_j(\theta)) - \eth_{\text{2D}}(\Pi_{\triangleleft}(\mathscr{J}_j(\theta)))\|^2 \tag{7}$$

where $\mathscr{J}_j(\theta)$ is the 3D position of joint j under pose θ.

4) limit term: This term aims to limit each individual joint to stay in a possible range of motion.

$$E_{\text{limit}}(\theta) = \frac{1}{N_\theta - 6} \sum_{i=1}^{N_\theta - 6} \big(\max(0, \psi_i^{\min} - \psi_i(\theta)) + \max(0, \psi_i(\theta) - \psi_i^{\max})\big)^2 \tag{8}$$

$\psi^{\min} \in \mathbb{R}^{N_\theta - 6}$ and $\psi^{\max} \in \mathbb{R}^{N_\theta - 6}$ are soft lower and upper bounds of $N_\theta - 6$ joint angle respectively. These boundaries are set according to our observation.

5) imu term: This term penalizes the difference between the current forearm pointing direction from the model and the observation from the wrist-worn IMUs.

$$E_{\text{imu}}(\theta) = \|\mathsf{C}_{\text{left}}(\theta) - \mathsf{P}_{\text{left}}\|^2 + \|\mathsf{C}_{\text{right}}(\theta) - \mathsf{P}_{\text{right}}\|^2 \tag{9}$$

$\mathsf{C}_{\text{left}}(\theta)$ and $\mathsf{C}_{\text{right}}(\theta)$ extract left and right forearm pointing direction as a 3D unit vector from the skeleton of pose θ

$$\mathsf{C}_{\text{left}}(\theta) = \text{II}(\mathscr{J}_{\text{leftWrist}}(\theta) - \mathscr{J}_{\text{leftElbow}}(\theta)) \tag{10}$$

$$\mathsf{C}_{\text{right}}(\theta) = \text{II}(\mathscr{J}_{\text{rightWrist}}(\theta) - \mathscr{J}_{\text{rightElbow}}(\theta)) \tag{11}$$

P_{left} and $\mathsf{P}_{\text{right}}$ are 3D unit vectors from left and right wrist-worn IMUs. As the IMU orientation in the camera reference frame is calculated, we can choose the direction of the IMU axis that points along the forearm pointing direction.

6) float term: The *float* term is introduced to deal with some parts of the mesh model that is currently projecting to the 2D silhouette area (no penalty from *bg* term) but is floating between the point cloud surface and the camera. That part of the model should be pushed to the closest spot in the volume behind the point cloud.

$$E_{\text{float}}(\theta) = \frac{1}{Q} \sum_{q=1}^{Q} \|S_p(\nu_q^{\text{float}}, \theta) - \eth_{\text{3D}}(S_p(\nu_q^{\text{float}}, \theta))\|^2 \tag{12}$$

$\mathcal{U}^{\text{float}} = \{\nu_q^{\text{float}}\}_{q=1}^{Q} \subseteq \mathcal{U}^{\text{bg}}$ is a set of selected points on the model. These points are inside the 2D body silhouette when projected but are shallower than the point cloud surface. This set will get updated every Levenberg-Marquardt iteration.

$\eth_{\text{3D}} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ maps a 3D position from the model to 3D position of the closest point in the volume behind the 3D point cloud surface. If the input is behind the point cloud surface, $\eth_{\text{3D}}$ will act as an identical function. This function is pre-estimated by discretizing the capture volume with a resolution of 0.02 m.

E. Vector of Residues

All energy terms can be converted to a sum-of-squares form in order to use Levenberg-Marquardt algorithm. By applying such conversion to all energy terms and given that $\Theta = (\theta, \mathcal{U})$, the total energy term (Eq.4) becomes

$$E(\Theta) = \sum_{k=1}^{K} r_k^2(\Theta) \tag{13}$$

$\mathbf{r}(\Theta)$ is a *vector of residues* which contains residual vectors from all energy terms.

$$\mathbf{r}(\Theta) = [\mathbf{r}_{\text{data}}(\Theta); \mathbf{r}_{\text{bg}}(\theta); \mathbf{r}_{\text{bgJoint}}(\theta); \mathbf{r}_{\text{float}}(\theta); \mathbf{r}_{\text{limit}}(\theta); \mathbf{r}_{\text{imu}}(\theta)]^\top \in \mathbb{R}^K \tag{14}$$

where $K = K_{\text{data}} + K_{\text{bg}} + K_{\text{bgJoint}} + K_{\text{float}} + K_{\text{limit}} + K_{\text{imu}}$ and $K_{\text{data}} = 6N_s$, $K_{\text{bg}} = 2H$, $K_{\text{bgJoint}} = 2|\mathcal{J}_{bg}|$, $K_{\text{float}} = 3Q$, $K_{\text{limit}} = N_\theta - 6$, $K_{\text{imu}} = 6$.

F. Levenberg-Marquardt (LM) Iteration

The goal is to find the pose parameter θ and the correspondence $\mathcal{U}$ that minimize the energy function.

$$\hat{\theta}, \hat{\mathcal{U}} = \underset{\theta, \mathcal{U}}{\text{argmin}}\, E(\theta, \mathcal{U}) \tag{15}$$

In each iteration, the LM algorithm tries to find an update $\Delta\Theta = (\Delta\mathcal{U}, \Delta\theta)$ that brings the current search to the lower energy state by forming the following equation system.

$$(J(\Theta)^\top J(\Theta) + \gamma I)\Delta\Theta = -J(\Theta)^\top \mathbf{r}(\Theta) \tag{16}$$

given that $J(\Theta)$ is the Jacobian of the residual vector $\mathbf{r}(\Theta)$.

The update $\Delta\Theta(\gamma)$ is solved as a function of the damping parameter γ and used to evaluate the new point. The damping parameter γ will be increased exponentially until $E(\Theta \oplus \Delta\Theta(\gamma'))$ becomes lower than $E(\Theta)$. Then, the update operation $\Theta \leftarrow \Theta \oplus \Delta\Theta(\gamma')$ is done to finish one iteration.

The update operator $\oplus$ for θ is an addition but the update operator for $\mathcal{U}$ is a mesh walking process (Section II-I).

G. Pose Initialization

Pose parameters are initialized from 2 sources. The first pose is the result from the previous frame. The second pose is built analytically to replicate the Kinect skeleton. Each starting point will be optimized independently on its own thread. The final pose with the lowest energy will be selected.

H. Sparse Sampling from Depth Pixels

Normally, the number of depth pixels in the silhouette can be as large as 50,000. Fitting a model to all available pixels cost excessive computation time. Therefore, they are subsampled to N_s pixels. A sampling strategy from Taylor *et al.*'s hand tracking [9] is adopted and modified to ensure 2 concerns within a sparser sampling. First, there should be enough samples from small body segments (*i.e.,* arm and head). Second, pixels from the near-edge area of the silhouette in every body segment must be sampled. Hence, the Kinect skeleton is utilized to categorize pixels into multiple zones. Each zone will be sampled according to an assigned quota to ensure a good distribution on the silhouette.

Each sampled pixel has its associated 3D position p_i, and its neighbor points usually form a 3D surface. We collect neighbor pixels that are within 5 units of Chebyshev distance in the pixel domain and within 0.05 m (Euclidean distance) in the camera space domain. Principal Component Analysis (PCA) is used to analyze those points in 3D space. The third principal component is selected as the normal vector on that sampled point. As a result, we get the 3D positions $\{\mathbf{p}_i|\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^{N_s}$ and the 3D normals $\{\mathbf{n}_i|\mathbf{n}_i \in \mathbb{R}^3; \|\mathbf{n}_i\| = 1\}_{i=1}^{N_s}$ associated to those sampled pixels.

I. Correspondence Continuous Update (Mesh walk)

When a correspondence point $\mathcal{U} = (f, u, v)$ has to be updated by $(\Delta u, \Delta v)$, the point could move beyond the boundary of the triangle face f to an adjacent face and so on. In that case, we adopt a mesh walking concept proposed by Taylor *et al*. [17], which tries to maintain the tangential continuity when the point has to transfer between 2 faces. The update repeats until the updated distance is finished.

J. Correspondence Discrete Update

Similar to the Taylor *et al*.'s method [9], in order to initialize correspondence $\mathcal{U}$ for the next iteration, the latest result from the mesh walking step will be compared to a set of candidates to choose the best correspondence for the i^{th} sampled point.

$$\nu_i \leftarrow \underset{\nu \in \{\nu_i\} \cup\, \mathcal{U}^{\text{cand}}_{t\%10}}{\operatorname{argmin}} \left(\frac{\|S_p(\nu, \theta) - p_i\|^2}{\sigma_p^2} + \frac{\|S_\perp(\nu, \theta) - n_i\|^2}{\sigma_n^2} \right) \tag{17}$$

$\mathcal{U}^{\text{cand}}_{t\%10} \subseteq \Omega$ is a pre-randomized set of correspondence candidates for iteration t. The random is designed to have 3 times more density on the upper limbs, neck, and head than the rest of the body.

As the surface area of the human model is much larger than the hand model, instead of increasing the number of the candidates in $\mathcal{U}^{\text{cand}}$ to cover the whole area, a new strategy is proposed to mitigate the problem while using a low number of candidates.

After an update using Eq.17, ν_i will be compared to 3 centroid positions from 3 adjacent faces. If one of those adjacent candidates gives a better match, ν_i will be updated to that point. This process will be repeated until ν_i stops updating. By this way, the ν_i will quickly move to a better point despite the sparse candidate distribution.

III. Model Personalization

A method is designed to search for a 10-dimensional shape vector (β) which matches the shape of the subject from one depth image frame with a controlled pose shown in Fig.3. The steps are a rough estimation followed by a fine tuning.

A. Rough Shape Estimation using Gradient Descent

This step searches for the shape vector (β) that provides the template model with bone lengths and sizes of body parts that are close to the lengths and sizes from Kinect skeleton tracking results. Pose parameters are ignored in this step. An error function is defined as

$$E_s(\beta) = \sum_{\ell \in \text{Segments}} (\mathcal{L}_\ell(\beta) - \mathcal{K}_\ell)^2 \tag{18}$$

$\mathcal{K}_\ell$ is the length of measurement ℓ from the Kinect skeleton. $\mathcal{L}_\ell(\beta)$ is the length of measurement ℓ from the template model with shape β.

Segments = {*upperArm*, *forearm*, *wholeLeg*, *hipToShoulder*, *shoulderToShoulder*, *ankleToHead*, *wholeArm*}. Each measurement can be described in the following details.

- *upperArm*: Average length of both upper arms.
- *forearm*: Average length of both forearms.
- *wholeLeg*: Sum of the average upper leg length (hip to knee) and the average lower leg length (knee to ankle).
- *hipToShoulder*: Distance from the average position of the hips to the average position of the shoulders.
- *shoulderToShoulder*: Distance from the left shoulder to the right shoulder.
- *ankleToHead*: Distance from the average position of the ankles to the head position (at the jaw height).
- *wholeArm*: Sum of the average upper arm length, the average forearm length, and the average hand length.

The shape vector β is initialized at the mean shape (a zero vector). A gradient descent method is used to iteratively update β by

$$\beta \leftarrow \beta - \lambda \nabla E_s(\beta) \tag{19}$$

$\nabla E_s(\beta)$ is the gradient of the error function (Eq.18) at point β calculated from an automatic differentiation library. The learning rate λ is initialized at 1. Whenever the update increases the error, λ will be reduced by half. The optimization is considered converged when λ is below 0.01. After that, β will be used to initialize the fine shape tuning step.

B. Fine Shape Tuning with Shape & Pose Fitting

The explained pose estimation method is modified to search for correspondences ($\mathcal{U}$), shape (β), and pose (θ) together as Θ_β. The same method is used to fit a 48-DOF full-body model ($N_\theta = 48$) to the full-body point cloud. As the pose is controlled, the energy terms we used are *data*, *bg*, *bgJoint*, and *limit* term.

IV. Evaluation

A. Gold Standard Measurement

To measure the accuracy, a Vicon motion capture system with 8 infrared cameras are used as a gold standard. The transformation between Vicon and Kinect reference frame is calibrated [13]. A synchronization between Kinect and Vicon is done via a LAN cable at the beginning of each record.

Plug-in Gait model for the upper body with 27 marker locations (from the hips and above) is chosen. In order to calculate shoulder, elbow, and wrist joint centers from those surface marker positions, shoulder offset, elbow width, and wrist thickness need to be measured from each subject. The detailed calculations of joint center calculation are described in a Plug-in Gait's document [18].

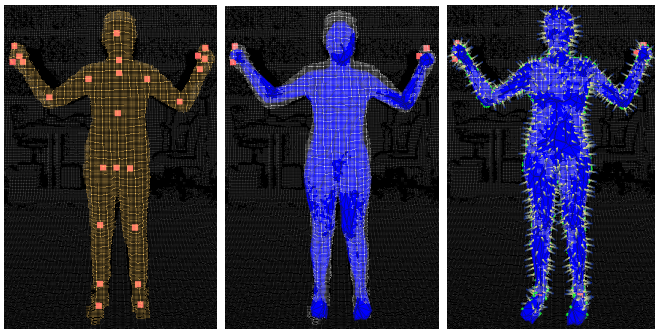


Fig. 2: A subject in the personalization process. Left: Full-body point cloud with Kinect's joint positions. Middle: Result from the rough shape estimation (Section III-A). Right: Result from the fine shape tuning (Section III-B).

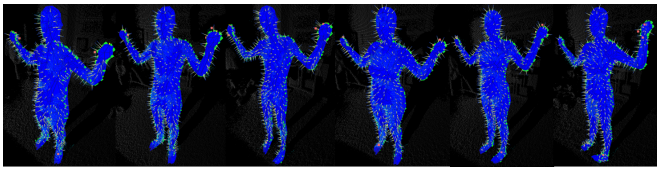


Fig. 3: Shape fitting results from 6 subjects in a controlled pose. Note that male (left half) and female (right half) use 2 different templates.

B. Subjects & Movement Sequences

Three healthy males and three healthy females are recruited as our subjects in this experiment. Each subject will move in 6 sequences. Some example movements are arm raising, shoulder lifting, torso tilting, air punching, forearm rolling, hair combing, jumping, etc. All the movements are done with both hands close as this study does not model the fine structure of hands and fingers. In total, 39,311 depth frames (about 22 minutes) are collected. All the recorded data are made publicly available [19].

C. Parameter Configurations

The configurations used in this evaluation are

- $N_s = 250$, $N_\theta = 40$, $\sigma_p = 0.003$, $\sigma_n = 0.25$
- $\lambda_{\text{bg}} = 10$, $\lambda_{\text{bgJoint}} = 25$, $\lambda_{\text{float}} = 0.25$
- $\lambda_{\text{limit}} = 10000$, $\lambda_{\text{imu}} = 15$, H_{max} = 150
- The maximum LM iteration allowed per frame is 20
- The initial damping parameter $\gamma = 1000$
- The moving average window size at the post-processing step covers 3 consecutive depth frames.

V. Results & Discussions

A. Model Personalization

Our personalization strategy can adjust the template model to the different sizes for both male and female subjects. The fitting results can be seen in Fig.2 and 3.

B. Quantitative Pose Estimation Results

All the position errors of the upper limbs (shoulder, elbow, and wrist) can be seen in Fig.4. They show 2 sets of result. The first set is from the whole dataset. The second set is from

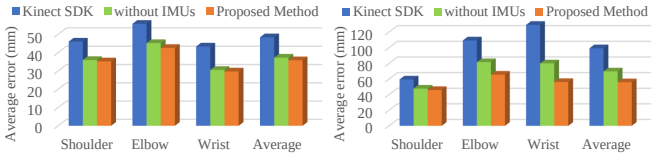


Fig. 4: Comparisons of joint position errors from full dataset (left) and from frames with occlusions (right). Our method reduces the average error from Kinect SDK by 25.9% in general cases and 43.7% in cases with occlusions.

1,452 selected frames with occlusions. In such occluding scenarios, dropping the *imu* term causes the wrist and elbow position error to increase 42.5% and 24.5% respectively.

C. Qualitative Improvement Over Kinect SDK

1) Flexible Spline: When a subject tilts his body to one side, Kinect SDK always has a problem in tracking joints along the body core. Neck, SpineShoulder, SpineMid, and SpineBase joint positions will form a strictly straight line which is incorrect. The Kinect skeleton also wrongly tracks an arm on one side as a consequence. On the other hand, the SMPL skeleton model has multiple DOF at the spine which allow the upper torso section to bend naturally and fit the model more correctly to the observation (Fig.5a).

2) Occluded Shoulder: When a shoulder is occluded by an arm section or by the torso, Kinect skeleton usually picks a shoulder position lower than it should (Fig.5b). Our method can correct such problem in most of the cases.

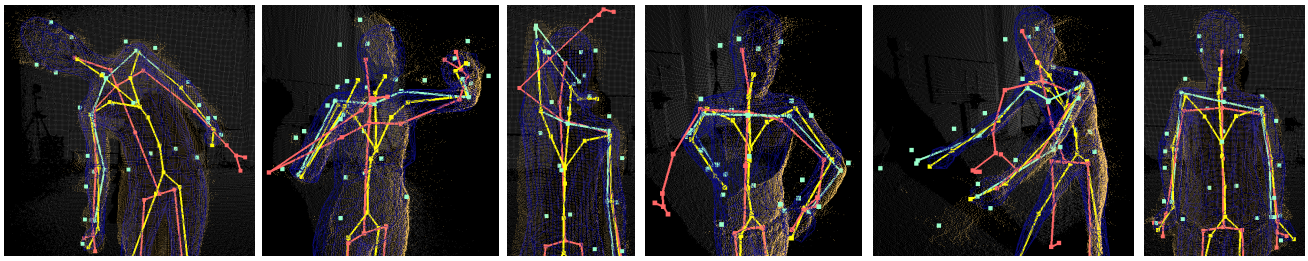
3) Occluded Arm: Occasionally, an arm can be partially or fully occluded by the other arm, by the head, or by the torso. In those cases, Kinect usually has notably less clue to infer the elbow and shoulder positions. However, the wrist-mounted IMUs give our method an access to the forearm pointing direction which often helps when the visual information is not complete. Fig.5c, 5d, and 5e show some of those cases seen in the dataset.

4) Arm Jumping in Noisy Frame: When forearms are between torso and the camera, the depth sensor is likely to generate a lot of *flying-pixel* noise at the edge of the forearms. It commonly confuses the Kinect skeleton for upper limb tracking. Our method usually mitigates such problem.

5) Shoulder Lift: When subjects lift up one or both of their shoulders, the movement range extracted from Kinect SDK will be diminished and become too subtle to be observed easily. Thanks to the design of SMPL model that does not stick shoulders rigidly to the torso, the shoulder can move around in a reasonable range and such movements can be captured as shown in Fig.5f. This feature opens a new opportunity to detect a common compensatory movement in upper-limb rehabilitation exercises without any markers.

D. Limitations & Future Works

A few issues about the correctness of the tracking have been observed. First, self-penetration still occurs in our tracking but not often. It can be avoided by adding a self-penetration penalty to the energy function. Second, as our



(a) (b) (c) (d) (e) (f)

Fig. 5: The cyan linkages are the joint positions from Vicon system (the gold standard). The pink linkages are the Kinect skeleton. The yellow linkages are from the proposed method. **(a)** When a subject tilt his body, notice that the Kinect spine is unnaturally straight. The spine from our method is more realistic. **(b)** When a shoulder is occluded, notice that shoulder positions of Kinect skeleton are off to the lower positions. **(c)** The right forearm is behind the head. **(d)** The right arm swings to the space behind the torso. **(e)** The right arm is heavily occluded by the left arm. **(f)** One shoulder is lifted.

method relies on the Kinect skeleton as an initialization pose of each frame, the tracking can simply go wrong if the subject turns more than 90 degrees away from the camera. However, such large turn can simply be excluded from the design of exercise prescription.

Currently, the algorithm takes about 1.5 seconds per frame on a CPU (Intel Core i7-3370). This speed limits our method to a group of applications which allows delayed feedbacks such as rehabilitation exercise assessment. However, a proper implementation of the method on a graphic processing unit (GPU) should be our next step to attend a real-time speed.

VI. Conclusion

With an upper-body rehabilitation exercise assessment as the main target application, a method to improve the Kinect skeleton tracking has been designed, implemented, and evaluated. Our key contributions are 1) the integration of the personalizable SMPL model with the simultaneous pose-correspondence optimization technique which allows the detection of scapular movements, 2) the demonstration of the benefit of adding IMU term directly to the generative pose optimizer in the context of a single depth camera, and 3) the release of the first public mocap dataset that contains the record from Kinect V2, synchronized IMUs, and a marker-based motion capture system [19]. The evaluation on the dataset shows both quantitative and quantitative improvements over the Kinect SDK. Some features such as the shoulder lift detection is found to be unique and could be used to enhance the automation of rehabilitation exercise assessments.

References

[1] F. Bogo, M. J. Black, M. Loper, and J. Romero, "Detailed full-body reconstructions of moving people from monocular rgb-d sequences," in *Proc. the IEEE Int. Conf. on Computer Vision*, 2015, pp. 2300–2308.

[2] M. Ye, Y. Shen, C. Du, Z. Pan, and R. Yang, "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1517–1532, Aug 2016.

[3] M. Ding and G. Fan, "Articulated and generalized gaussian kernel correlation for human pose estimation," *IEEE Trans. Image Process*, vol. 25, no. 2, pp. 776–789, Feb 2016.

[4] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, "Accurate 3d pose estimation from a single depth image," in *2011 Int. Conf. on Computer Vision*, Nov 2011, pp. 731–738.

[5] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*. Springer London, 2013.

[6] T. Helten, M. Mller, H. P. Seidel, and C. Theobalt, "Real-time body tracking with one depth camera and inertial sensors," in *2013 IEEE Int. Conf. on Computer Vision*, Dec 2013, pp. 1105–1112.

[7] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, "Efficient human pose estimation from single depth images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 2821–2840, 2013.

[8] Q. Wang, G. Kurillo, F. Ofli, and R. Bajcsy, "Evaluation of pose tracking accuracy in the first and second generations of microsoft kinect," in *2015 Int. Conf. Healthcare Inform.*, 2015, pp. 380–389.

[9] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, *et al.*, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," in *ACM SIGGRAPH Conf. Comput. Graphics and Interactive Techn.*, June 2016.

[10] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[11] G. Pons-Moll, A. Baak, T. Helten, M. Mller, H. P. Seidel, and B. Rosenhahn, "Multisensor-fusion for 3d full-body human motion capture," in *2010 IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognition*, June 2010, pp. 663–670.

[12] T. Helten, A. Baak, G. Bharaj, M. Mller, H. P. Seidel, and C. Theobalt, "Personalization and evaluation of a real-time depth-based full body tracker," in *2013 Int. Conf. on 3D Vision*, June 2013, pp. 279–286.

[13] P. Jatesiktat and W. T. Ang, "Recovery of forearm occluded trajectory in kinect using a wrist-mounted inertial measurement unit," in *2017 39th A. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, pp. 807–812.

[14] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "SCAPE: shape completion and animation of people," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 408–416, 2005.

[15] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. 24th A. Conf. Comput. Graphics and Interactive Techn.*, ser. SIGGRAPH '97, 1997, pp. 209–216.

[16] D. Ebery. (2016, May) Mesh differential geometry. [Online]. Available: https://www.geometrictools.com/Documentation/MeshDifferentialGeometry.pdf#page=9

[17] J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, and A. Fitzgibbon, "User-specific hand modeling from monocular depth sequences," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[18] (2016) Plug-in gait. [Online]. Available: http://www.irc-web.co.jp/vicon_web/news_bn/PIGManualver1.pdf#page=42

[19] P. Jatesiktat. NTU motion capture dataset. [Online]. Available: https://koonyook.github.io/mocap_dataset/