Visual-Inertial Filtering for Human Walking Quantification

Marc Mitjans\(^1\)\(^*\), Michail Theofanidis\(^1\)\(^*\), Ashley N. Collimore\(^2\), Madelaine L. Disney\(^3\), David M. Levine\(^3\), Louis N. Awad\(^2\), Roberto Tron\(^1\)

Abstract—We propose a novel system to track human lower-body motion as part of a larger movement assessment system for clinical evaluation. Our system combines multiple wearable Inertial Measurement Unit (IMU) sensors and a single external RGB-D camera. We use a factor graph with a Sliding Window Filter (SWF) formulation that merges 2-D joint data extracted from the RGB images via a Deep Neural Network, raw depth information, raw IMU gyroscope readings, and estimated foot contacts extracted from IMU gyroscope and accelerometer data. For the system, we use an articulated model of human body motion based on differential manifolds. We compare the results of our system against a gold-standard motion capture system and a vision-only alternative. Our proposed system qualitatively presents smoother 3D joint trajectories when compared to noisy depth data, allowing for more realistic gait estimations. At the same time, with respect to the vision-only baseline, it improves the median of the joint trajectories by around \(0\) cm, while considerably reducing outliers by up to \(0.6\) cm.

I. INTRODUCTION

Frail older adults constitute a fast-growing segment of the population. Many aged individuals suffer from multiple disorders associated with the loss or impairment of motor functions [1]. Modern health care systems fail on most accounts to detect the changes in a person’s health status that often precede catastrophic events, thus leading to intensive, high-cost, and institution-based interventions that may have been avoided with better monitoring. Changes in physical activity and movement dysfunctions are often the first indicators of frailty. Home-based technologies that continuously assess real world mobility have high potential to facilitate early detection and treatments. In this light, we propose a multimodal movement estimation system that could be used for home-based mobility monitoring, and ultimately for the timely diagnosis of movement impairments.

Fig. 1 illustrates the overall architecture of the proposed system. In the current instance, we use a single stationary 3D depth-sensing camera and four Inertial Measurement Unit (IMU) sensors strapped on the lower body of the subject. We use a pre-trained Deep Neural Network (DNN) model (OpenPose [2]) to extract 2-D joint data from the RGB images, and a Logistic Regressor (LR) classifier to estimate foot contacts. These computed measurements are then combined via a factor graph with raw depth and IMU gyroscope measurements, based on a differential manifold representation of an articulated human model; we use the Georgia Tech Smoothing and Mapping (GTSAM) [3] for inference on the factor graph.

Related Work. There is substantial interest in measuring human motion using vision and inertial measurement approaches. For example, the authors of [4] track the upper limb of patients undergoing rehabilitation with the use of two IMU sensors and a kinematic model. The authors of [5] and [6] similarly use a dynamic and kinematic model to estimate the pose of a pedestrian from a single RGB camera. Furthermore, the authors of [7] use IMU sensors to track the human body with the aid of a kinematic model from a Kinect sensor. Lastly, several algorithms that require RGB-D data [8], [9] such as of [10] and [11] propose multi-segment tracking algorithms that decomposed RGB-D data volumetrically into regions that represent the human body skeleton. In comparison to these studies, the proposed system uses a tighter integration of machine learning and a filtering framework with an additional foot contact detector. This improves our system with the capabilities of machine learning algorithms, while still benefiting from the robustness of model based approaches.

In parallel, the fusion of vision and inertial measurements represents a cornerstone of state estimation for robots, typically referred to as Visual-Inertial Odometry (VIO). Approaches in this domain can be divided into two categories:
loosely coupled and tightly coupled [12], [13]. Loosely coupled approaches use the IMU for the update step of the estimation together with simple models (e.g., constant velocity) for the prediction. On the other hand, tightly coupled approaches use the IMU data directly in the prediction step, without requiring a priori models for the system dynamics. Virtually all recent VIO approaches are based on a tightly coupled architecture. These approaches can be further divided into two categories: filtering methods and nonlinear optimization methods. The most representative work for filtering methods is the Multi-State Constraint Kalman Filter (MSCKF) [14], [15], which expands upon the Extended Kalman Filter (EKF). Historically, filtering methods were mainly used for their low computational requirements, which easily allowed real-time performance. More recently, Sliding Window Filtering (SWF) approaches based on nonlinear optimization via factor graphs [16], [17] have emerged as the state of the art paradigm; these methods perform maximum likelihood estimation on a limited sliding window of measurements. Although computationally more intensive, they provide better performance [18].

**Paper contributions.** Our system applies recent advancements from the VIO literature to the domain of human motion estimation, showing that it is possible to augment the raw measurements with the outputs of machine learning algorithms, and that such outputs can be successfully integrated in factor graph methods. We report performance of the system when used as part of clinically-relevant mobility testing.

**II. PRELIMINARIES**

In this section we cover the mathematical background and preliminaries required to implement the proposed SWF for 3D human walker tracking. For additional details, we refer to existing literature on Lie groups [19], [20], optimization on manifolds [21], and factor graphs [3].

**A. The rotation group SO(3)**

The space of 3-D rotations is defined as $SO(3) = \{ R \in \mathbb{R}^{3 \times 3} : R^T R = I_3 \}$ and is a Lie group (i.e., it has a group structure given by the matrix multiplication and is also a differential manifold). Its tangent space at the identity, the Lie algebra $so(3)$, is given by the space of $3 \times 3$ skew symmetric matrices; it can be identified with $\mathbb{R}^3$ via the *hat* operator, which maps a vector $\omega \in \mathbb{R}^3$ to a matrix $\omega^\wedge \in so(3)$ encoding the cross product, i.e., $\omega \wedge v = \omega \times v$ for any $v \in \mathbb{R}^3$. Tangents to a time-varying curve $R$ can be written in the form of $\dot{R} = R \dot{\omega}$, where $\dot{\omega}$ is the time derivative of $\omega$. The exponential map at the identity, denoted by $exp : so(3) \rightarrow SO(3)$, maps an element $\omega \in so(3)$ to a rotation $R$ along the geodesic (under the standard Riemannian metric on $SO(3)$) in the direction $\omega$. The logarithm map $log : SO(3) \rightarrow so(3)$ is locally defined as the inverse of the exponential map. The *left* and *right Jacobian* of the exponential and logarithm maps give the relation between tangents of curves in $so(3)$ and tangents of curves on $SO(3)$, and can be computed in closed form (see [22] for details).

**B. Optimization on manifolds**

Let $\mathcal{M}$ be a Riemannian manifold. A *retraction* $R_x$ on $\mathcal{M}$ is a map from the tangent space at $x$ to the manifold (the exponential map $exp$ defined in Sec. II-A is an example).

Let us consider $\min_{x \in \mathcal{M}} g(x)$ as an optimization problem over $\mathcal{M}$. The cost function $g(x)$ can be lifted [23] via a retraction to the tangent space, i.e., $g(\mathcal{X})$ can be transformed to $\bar{g}(R_{\mathcal{X}}(\delta \mathcal{X}))$, where $\delta \mathcal{X}$ is an element of the tangent space. Algorithms for optimization on manifolds then follow a two-step process: 1) use the lifted cost function to determine a direction $\delta \mathcal{X}$ that locally decreases the cost; 2) use the retraction to move $\mathcal{X}$ to $R(\mathcal{X})$. The first step typically requires computing the gradient of the lifted cost $\bar{g}$, which in turn requires the Jacobian of the retraction.

**C. Factor graphs**

A factor graph $G = (\mathcal{S}, \mathcal{F})$ is a bipartite graphical model that represents a probability distribution given by the multiplication of factors $\mathcal{F} = \{ f_k \}$ that are functions of subsets $s_k$ of the states $\mathcal{S} = \bigcup_k s_k$ [24]; typically, each factor will depend also on a subset $\mathcal{Z}_k$ of external measurements $\mathcal{Z} = \{ \mathcal{Z}_k \}$. Factor graphs are at the base of the current state-of-the-art results in Simultaneous Location And Mapping (SLAM) [25]. More in detail, the likelihood of $\mathcal{S}$ given $\mathcal{Z}$ is defined as from the factors $\mathcal{F}$ as

$$P(\mathcal{S}|\mathcal{Z}) \propto \prod_k f_k(s_k; \mathcal{Z}_k).$$

The maximum likelihood (ML) estimate of $\mathcal{S}$ corresponds to maximizing (1), or equivalently, minimizing its negative log-likelihood. As it is common in the literature [3], [26], we assume that each factor $f_k$ is given by residuals $r_k$ that follow a zero-mean Gaussian distributions with a covariance $\Sigma_k$. The final optimization problem associated to $G$ can then be written as a nonlinear least-squares optimization problem:

$$S^* = \arg \min_S -\ln \prod_k f_k(S) = \arg \min_S \sum_k \| r_k(S) \|_2^2.$$
III. PROBLEM FORMULATION

A. Human walker model

We represent a human walker by a tree $\mathcal{H} = (\mathcal{V}, \mathcal{E})$. The nodes $n^i \in \mathbb{R}^3$, $i \in \mathcal{V}$ correspond to 3-D joint positions expressed in a fixed inertial frame, and with $i = 0$ representing the root node). The edges $(i, j) \in \mathcal{E}$ correspond to links; for each one, we associate a length $l_{ij}^i \in \mathbb{R}$ that is independent of $t_k$, and a rotation $R_{ij}^i \in SO(3)$ for each time $t_k$, defined as a rotation from the link to the fixed inertial frame with its $z$ axis pointing downward along the direction of the link. See Fig. 2 for a graphical depiction of the model.

The kinematic relation for each edge $(i, j) \in \mathcal{E}$ is:

$$n_{ij}^k = n_k^i + l_{ij}^i R_{ij}^i \epsilon_3 = n_k^i + l_{ij}^i \exp(r_{ij}^i) \epsilon_3,$$

where $r_{ij}^i = \log(R_{ij}^i)$, and $\epsilon_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ corresponds to the standard basis $z$ axis coordinates. Our model parametrization (in contrast, e.g., to a naive Cartesian joint representation), allows us to separate extrinsic quantities (rotations) from intrinsic model parameters (lengths).

B. Measurements

We capture temporal data from Inertial Measurement Units (IMUs), and spatial data from an RGB-D camera, from which we extract four different types of measurements. The time instants $t_k$ are given by the camera time stamps.

1) Inertial Measurement Units: We assume that the average angular velocities for four links (two thighs and two shanks) are measured by IMUs placed in the middle of each link (see Fig. 2). We assume a consistent placement of the IMUs with respect to the body, so that the measurements can be expressed in the correct reference frame via a fixed rotation. Since IMU measurements are received at a higher rate than image-based measurements, we use preintegration [17] to combine multiple IMU measurements into a single relative rotation $\Delta \tilde{R}_{ij}^k$ such that $\Delta \tilde{R}_{ij}^k = R_{ij}^{t_k} - R_{ij}^{t_{k-1}}$ in the absence of noise. We denote as $Z_k = \{\Delta \tilde{R}_{ij}^k\}$ the full set of preintegrated IMU rotations at time $t_k$.

2) RGB images: Images are processed using the OpenPose DNN model [2], which outputs 2-D image plane pixel coordinates $\tilde{x}_k^i, \tilde{y}_k^i$ for each joint $i \in \mathcal{V}$ in our human model and for each frame $t_k$. Note that OpenPose uses a cascade of regressors that results in holistic estimations of the joint position, i.e., the coordinates of hidden joints are sometimes inferred from the detections of the visible joints. We denote as $P_k$ the full set of coordinates for all the joints.

3) Depths: The RGB-D camera provides depth data frames synchronously with the RGB images at each time $t_k$. Using a known camera calibration matrix $K$, we back-project the 2-D coordinates from the OpenPose measurements $P_k$ measurements to the point cloud to recover the depth $z_k^i$ for each joint. We denote as $D_k$ the full set of depths for all the joints at time $t_k$. Note that an alternative approach would be to combine image and depth measurements into a single 3-D joint estimate. However, this would couple two modalities that have very different noise models, making the determination of appropriate covariance matrices difficult.

4) Contacts: We use IMU data to extract additional foot contact measurements, which are used to introduce factors that fix the position of feet that are on the ground. To avoid the use of additional sensors for step detection (as in [16], [27]), some of which would require more cumbersome setups for the human subject, we produce estimates of the contact state between the feet and the ground by using a trained binary classifier on the angular velocity and acceleration data from the IMUs in Sec. III-B.1; specifically, we use input feature vectors in $\mathbb{R}^{120}$ obtained by concatenating (non-preintegrated) gyroscope and accelerometer data over 30 successive measurements. For the outputs, we manually labelled 2722 keyframes in 18 datasets independent from the ones presented in Sec. V. We trained the classifier using 70% of the data for training, and 30% for testing the performance of the classifier. We tested both a Recurrent Neural Network (RNN) and a Logistic Regression (LR) model, and while both gave similar results, the latter was finally chosen due to its simplicity. To reduce the influence of outliers, the output of the LR classifier is further processed with a median filter. In the end, our classifier showed a 95% classification accuracy. We denote as $C_k$ the set of contact estimations at time $t_k$.

IV. FACTOR GRAPH

We define the state $R_k$ of the human walker at time $t_k$ by concatenating the root node position $n_0^k$ with all rotations $R_{ij}^k$ to form state, and we also define a calibration state $L$ containing all the lengths $l_{ij}^i$. As previously mentioned, we define the factor graph on a window of $n$ states $S_k = \{ L, \{ R_{ij}^k \}_{i=k-n+1}^{k-1} \}$ over the time interval $(t_{k-n}, t_k]$. The corresponding set of measurements is defined as $Z_k = \{ Z_k, P_k, D_k, C_k \}_{i=k-n+1}^{k-1}$. Note that we have one less measurement vector for $Z_k$ and $C_k$, as they are in-between keyframe measurements.
Fig. 3: A small window of the proposed factor graph. Connecting the orientation states \( \mathcal{R} \) and length state \( \mathcal{L} \), it shows the IMU (red), image plane (blue), depth (green), and contact (yellow) factors.

Our goal is to compute the ML estimate of the states \( \mathcal{S}_k \) given the measurements \( Z_k \). By assuming statistically independent measurements, (2) can be used to formulate our problem with the factor graph shown in Fig 3, leading to the following optimization problem:

\[
\mathcal{S}_k = \text{argmin}_{\mathcal{S}_k} \sum_{k'=k-n}^{k-1} \| \mathbf{r}_{f, k'} \|^2 + \sum_{k'=k-n}^{k-1} \| \mathbf{r}_{r, k'} \|^2 + \sum_{k'=k-n}^{k-1} \| \mathbf{r}_{c, k'} \|^2 \tag{4}
\]

where \( \mathbf{r}_{f, k'} \), \( \mathbf{r}_{r, k'} \), and \( \mathbf{r}_{c, k'} \) are the residuals (covariance matrices) of the IMU, image plane, depth and contact factors, respectively.

In the remainder of this section, we give expressions for the residuals and their Jacobian matrices, such that (4) can be solved using GTSAM (see Sec. II-B). We use superscripts \( i \) and \( ij \) to denote quantities that refer to a singular joint or link (full expressions are obtained by stacking these quantities).

A. IMU factor

The IMU factor computes the rotation error of each link between two consecutive keyframes based on the residual

\[
\mathbf{r}_{f, k}^{ij} = \log \left( (\mathbf{\Delta} \mathbf{R})^T_k (\mathbf{R}^{ij}_{k-1})^T \mathbf{R}^{ij}_k \right) \tag{5}
\]

Note that we do not model IMU bias in our factor graph. The factor above uses only gyroscope data, for which we empirically determined that the bias is negligible. Moreover, the accelerometer data is only used for contact detection, for which bias does not produce a significant difference.

1) Jacobians: The Jacobians of \( \mathbf{r}_{f, k}^{ij} \) with respect to the Lie representation of the rotations, \( \mathbf{r}_{k-1}^{ij} \) and \( \mathbf{r}_{k}^{ij} \), are:

\[
\mathbf{J}_{\mathbf{r}_{f, k}^{ij}}(\mathbf{r}_{k-1}^{ij}) = -\mathbf{J}_r^{-1}(\mathbf{r}_{k-1}^{ij}) \cdot (\mathbf{R}^{ij}_k)^T \mathbf{R}^{ij}_{k-1} \tag{6}
\]

\[
\mathbf{J}_{\mathbf{r}_{f, k}^{ij}}(\mathbf{r}_{k}^{ij}) = \mathbf{J}_r^{-1}(\mathbf{r}_{k}^{ij}) \tag{7}
\]

where \( \mathbf{J}_r^{-1}(\cdot) \) is the right Jacobian of the log map.

2) Covariance: The factor covariance for each link is given by the corresponding preintegrated IMU covariance; we refer to [17] for detailed expressions.

B. Image plane factor

The image plane factor computes a residual in 2-D metric coordinates:

\[
\mathbf{r}_{ip, k}^j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left( \begin{bmatrix} x_k^j \\ y_k^j \\ 1 \end{bmatrix} - K^{-1} \begin{bmatrix} x_k^j \\ y_k^j \\ 1 \end{bmatrix} \right) \tag{8}
\]

where \( x_k^j, y_k^j, z_k^j \) are the 3-D coordinates of the joint \( n^j \) in \( S \), and the camera calibration matrix \( K \) is used to transform the pixel coordinates from OpenPose to metric coordinates.

1) Jacobians: To compute the Jacobians with respect to parent rotations \( R_{k}^{pq} \) and link lengths \( l_{pq} \), we start from the kinematic relation (3):

\[
\begin{bmatrix} x_k^j \\ y_k^j \\ z_k^j \end{bmatrix} = \begin{bmatrix} x^0 + \sum_{(p,q)} [R_{k}^{pq}]_{02} l_{pq} \left[ (R_{k}^{pq})_{02} z_j^p \right] \\ y^0 + \sum_{(p,q)} [R_{k}^{pq}]_{12} l_{pq} \left[ (R_{k}^{pq})_{12} z_j^p \right] \\ z^0 + \sum_{(p,q)} [R_{k}^{pq}]_{22} l_{pq} \left[ (R_{k}^{pq})_{22} z_j^p \right] \end{bmatrix} \tag{9}
\]

where we omitted the subscript \( k \) for the sake of readability, \([R]_{ab}\) denotes the \( a, b \)-th element of the matrix \( R \), and the summations run over the the links \((p, q)\) in the kinematic chain from the root to \( j \). Then, for a given joint \( n^j \) we can compute its Jacobians with respect to a parent link \((p, q)\). The Jacobian for the link length \( l_{pq} \) is

\[
\mathbf{J}_{l_{pq}}^j = \begin{bmatrix} \frac{\partial (x^j / z^j)}{\partial l_{pq}} \\ \frac{\partial (y^j / z^j)}{\partial l_{pq}} \end{bmatrix} = \frac{1}{z^j} \begin{bmatrix} [R_{k}^{pq}]_{02} - x^j z_j^p [R_{k}^{pq}]_{22} \\ [R_{k}^{pq}]_{12} - y^j z_j^p [R_{k}^{pq}]_{22} \end{bmatrix} \tag{10}
\]

Similarly, for the link orientation \( r_{pq} \) we have:

\[
\mathbf{J}_{r_{pq}}^j = \begin{bmatrix} \frac{\partial (x^j / z^j)}{\partial r_{pq}} \\ \frac{\partial (y^j / z^j)}{\partial r_{pq}} \end{bmatrix} = \frac{l_{pq}}{z^j} \begin{bmatrix} e_{1j}^T J_{pq}^j - x^j z_j^p e_{3j}^T J_{pq}^j \\ e_{2j}^T J_{pq}^j - y^j z_j^p e_{3j}^T J_{pq}^j \end{bmatrix}, \tag{11}
\]

where \( J_{pq}^j = [-(R_{k}^{pq})_{21}, (R_{k}^{pq})_{12}, 0_{4 \times 1}] \in \mathbb{R}^{3 \times 3} \).

Finally, the Jacobian for joint \( n^j \) with respect to the system variable \( n^0 \) corresponds to:

\[
\mathbf{J}_{n^0}^j = \begin{bmatrix} 1 \\ 0 \\ \frac{-x^j}{(z^j)^2} \\ \frac{y^j}{(z^j)^2} \\ \frac{1}{z^j} \end{bmatrix} \tag{12}
\]

2) Covariance: Since it is difficult to obtain a principled quantification of the covariance in the OpenPose estimates, we treat \( \Sigma_{r_{pq}} \) as a design parameter. The fine-tuned parameter to present the results in Sec. V is \( \Sigma_{r_{pq}} = 7 \cdot 10^{-4} I_2 \).

C. Depth factor

The residual for the depth factor is simply defined as

\[
\mathbf{r}_{d, k}^j = z_k^j - \tilde{z}_k^j \tag{13}
\]
1) Jacobians: The Jacobian with respect to a parent link length $l_{pq}$ is trivial, and can be directly extracted from (9):

$$ J_{l_{pq}} = e_3^T R_{pq} e_3 $$

(14)

By following a reasoning similar to the previous factor, $J_{r_{pq}}$ and $J_{n_{pq}}$ are obtained with:

$$ J_{r_{pq}} = I_{pq} e_3^T J_{pq} $$

(15)

$$ J_{n_{pq}} = e_3^T $$

(16)

where $J_{pq}$ is computed in a similar manner as for (11).

2) Covariance: We experimentally observed that the noise variance in the depth measurements increases with the distance to camera. We therefore define a depth-dependent covariance matrix as follows. We captured a sequence of images in which a subject stood still in front of the camera at different distances, and captured the depth of all the joints as described in Sec. III-B.3. Assuming that joints at the same distance are identically distributed, we fit a logistic model $\sigma(\bar{z})$ mapping the sample joint depth mean to the sample standard deviation. The covariance of the depth factor is then obtained from an empirically re-scaled version of $\sigma$ with a tuning factor used to balance the scaling with other factors:

$$ \Sigma_{\Delta z_k} = 0.02 \sigma(\bar{z})^2 = 0.02 \left( \frac{1}{1 + \exp(-\bar{z} - 4)} \right)^2 $$

(17)

D. Contact factor

Contact measurements $C_k \in \{0, 1\}$ indicate whether the right foot (0) or the left foot (1) is in contact with the ground. This information is very useful to prevent unrealistic jittering of the estimated model along the $z$ axis (especially at larger distances, where the depth data is more noisy), and to reduce the influence of misdetection errors from OpenPose.

The residual for the factor is derived from a simple constant-position model on the coordinates of the selected foot, $n_{foot}$:

$$ r_{c_k} = r_{foot_{k+1}} - r_{foot_k} $$

(18)

1) Jacobians: Similarly to Sec. IV-C, we can compute the required Jacobians by recovering $J_{pq}$ from (11):

$$ J_{l_{pq}}^{foot} = R_{pq}^{foot} e_3^T - R_{pq} e_3^T $$

(19)

$$ J_{r_{pq}}^{foot} = I_{pq} J_{pq}^{foot} $$

(20)

$$ J_{n_{pq}}^{foot} = -R_{pq} J_{pq} $$

(21)

$$ J_{n_{pq}}^{foot} = -J_{n_{pq}}^{foot} = I_3 $$

(22)

2) Covariance: Although the synthetic contact measurements are obtained from IMU data, modeling their statistical correlation with the preintegrated rotation measurements is complex; hence, we opt to make the approximation that $C_k$ is statistically independent from $\Delta z_k$. Moreover, as in the case of image plane measurements, we treat the choice of the covariance as a tuning parameter for the model. For the experiments, we selected $\Sigma_{\Delta z_k} = 10^{-3} I_3$.

V. EXPERIMENTAL STUDY

A. Experimental Setup

We evaluate the proposed system on healthy human subjects using recordings of the 10-meter walk test (10MWT), a common clinical test consisting of an acceleration phase, a constant speed phase, and a deceleration phase. Fig. 4a shows the physical recording set up. The participant is outfitted with the four IMU sensors of our system (thighs and shanks), in addition to motion capture markers (ankles, knees, and hips) for recording ground-truth trajectories (see Fig. 4b for a diagram of the sensor placement). The RGB-D camera is placed longitudinally in the direction of the walking lane (Fig. 4a gives a sample RGB image).

The motion capture data are recorded by an 18-camera Qualysis Oqus-7 camera system at 200 hz. The RGB-D camera used is a RealSense d435 camera (Intel) capturing data at 30 fps. The four Xsens MTW-Awinda IMU sensors captured data at 120 hz. The data from our system is managed via the Robotic Operating System (ROS) [28] middleware. The extrinsic calibration (i.e., rigid pose transformation) between the Qualysis and the RGB-D camera is performed before every recording session by collecting data on a calibration target with a fiducial (AprilTag [29]) aligned with motion capture markers, followed by a standard 3-D to 3-D alignment procedure; the temporal calibration is performed through communication between the proprietary Qualisys recording software and ROS.

B. Analysis of Results

In total, we collected data from five healthy participants under the IRB protocol Mass General Brigham 2020P003474, with each participant completing six trials of the 10MWT. In three 10MWT trials, the participant walked toward the RGB-D camera; during the other three trials, the participant walked away from the camera. Fig. 4a provides a graphical illustration of the experimental setup by showing a trial with a participant performing the 10MWT by walking away from the camera. During each trial, we record six Cartesian trajectories from the marker data that represent the motion of the right and left hip, knee, and ankle joints. The system then computes an
initial estimation of these trajectories by projecting the pose information of OpenPose to the cloud data. Furthermore, the initial Cartesian trajectories are fused with the measurements of the IMU sensor by utilizing GTSAM to provide final estimations of the six captured trajectories.

Fig. 5a shows a sample trial with selected keyframes from the trajectories of the hips, knees and feet estimated by OpenPose and GTSAM with a 150-SWF. Note that, as the participant walks away from the camera, the estimations of OpenPose become more inaccurate due to the increased noise from the depth sensor. Additionally, our estimation produces more plausible gait trajectories. To compare quantitatively the performance of GTSAM and OpenPose, we calculated how close to the ground truth the estimations of GTSAM and standalone OpenPose are. For every joint of the skeleton in every keyframe, we compute the Euclidean distances of GTSAM and OpenPose with respect to the ground truth data. Fig. 5b presents a box plot that illustrates the four quartiles for the mentioned Euclidean joint errors in the trial represented in Fig. 5a. We can see that GTSAM with a 150-SWF reduces the outliers of OpenPose by more than 30 cm, and the values of the 75th percentile and median by a few centimeters.

Fig. 6 presents the Euclidean errors of OpenPose and GTSAM with two SWFs, of sizes 150 and 50, compared to Qualysis data for all the joint trajectories in all the trials. Again, for a SWF of size 150 we notice a similar pattern than in Fig. 5, with GTSAM reducing the maximum outliers with respect to OpenPose by about 0.6 m, and the median by a few centimeters. Fig. 6 also provides a third boxplot for a SWF of size 50. This time, although GTSAM removes the outliers of OpenPose, its overall performance decreased. This phenomenon can be attributed to the fact that an increased number of states allows retaining more past information than a filter with reduced size. A large filter size, however, comes with the added cost of increased computational complexity, which is an important factor when considering real-life applications. On average, the factor graph optimization for windows of size 50 took 0.22 s, while the optimization for windows of size 150 took 0.53 s.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a system that uses a human model and factor graph inference to fuse skeleton information from OpenPose with IMU data to estimate lower body human motion. The novelty of our work lies in the application of GTSAM (which is traditionally employed for localization and mapping in robots) to the domain of motion estimation for human walking, allowing the use of a human model based on manifolds, and the fusion of different sensor modalities with the output of different machine learning algorithms. In our experiments, we show that a sufficiently large Sliding Window Filter based on factor graphs qualitatively and quantitatively improves the 3D pose estimations with respect to a vision-only approach. In the future, we plan to investigate the use of our system in a home setting (where the use of the motion capture gold standard is not feasible), as well as the integration of the factor graph in the learning of the machine learning models, and the use of our system for automatic activity-based clinical assessment.
REFERENCES


