Robust Human Pose Tracking For Realistic Service Robot Applications

Manolis Vasileiadis[1,2], Sotiris Malassiotis[2], Dimitrios Giakoumis[2],
Christos-Savvas Bouganis[1], Dimitrios Tzovaras[2]

[1]Department of Electrical and Electronic Engineering, Imperial College London, UK
[2]Information Technologies Institute, CERTH, Greece

Abstract

Robust human pose estimation and tracking plays an integral role in assistive service robot applications, as it provides information regarding the body pose and motion of the user in a scene. Even though current solutions provide high-accuracy results in controlled environments, they fail to successfully deal with problems encountered under real-life situations such as tracking initialization and failure, body part intersection, large object handling and partial-view body-part tracking. This paper presents a framework tailored for deployment under real-life situations addressing the above limitations. The framework is based on the articulated 3D-SDF data representation model, and has been extended with complementary mechanisms for addressing the above challenges. Extensive evaluation on public datasets demonstrates the framework's state-of-the-art performance, while experimental results on a challenging realistic human motion dataset exhibit its robustness in real life scenarios.

1. Introduction

Human pose estimation and tracking refers to the process of detecting and extracting the positions of the joints of the human body from, either single or sequences of, RGB and depth images or 3D point-clouds, in order to reconstruct the skeletal structure and provide information about body posture, body motion and human gestures. It is considered one of the major challenges in the field of Computer Vision and has been intensively studied in the last few decades by the computer vision community [23, 22], due to its fundamental importance in various scientific fields. Pose estimation and tracking techniques have found usage in a large variety of technology domains, such as healthcare and robotics, with robust pose tracking becoming a basic pre-requisite for assistive service robots aiming towards monitoring human activities and providing assistance in daily life [19, 21].

Estimating the human pose is an intricate and complex task. The human body presents high variability in shape, size and texture, while the articulated joints that make up the human skeleton offer many degrees of freedom, providing a large range of motion for each rigid body part. Marker-based motion capture systems have been effectively used for body pose estimation and tracking, in controlled laboratory environments [27]. However, the intricate installation process and high cost have prevented the wide adoption of such systems in real-life applications. As a result, significant research focus has been put towards marker-less techniques, using consumer-grade RGB and depth cameras [22]. While most state-of-the-art techniques are reported to achieve high joint estimation accuracy and real-time performance, they are usually evaluated using datasets captured under ideal recording conditions in controlled laboratory environments (SMMC-10 [15], EVAL [16], PDT [17] datasets). Their accuracy and robustness, however, degrade in real life applications, where various problems arise, such as body part occlusions due to the presence of obstacles, partial-body views due to constraints in the available FOV of the camera, sensor noise, interaction with objects etc [26, 29].

The current work aspires to improve the robustness of human motion analysis in realistic settings, by introducing a real-time human pose estimation and tracking framework, which builds upon the articulated SDF-based model presented in [32] and is extended through a series of complementary features and mechanisms. The features proposed herein target specific problems encountered under real-life monitoring conditions, eventually leading to the development of a complete standalone framework suitable for deployment in real life, assistive robot applications.

The rest of the paper is organized as follows. Section 2 provides a summary of the state-of-the-art in the field of human pose estimation and tracking. Sections 3 and 4 present the base of the proposed framework's articulated tracker along with its complementary tracking features. Sections 5 and 6 describe the framework's initialization and data preprocessing steps respectively, Section 7 provides results from the experimental evaluation of the proposed framework and presents a new realistic human motion dataset and finally, Section 8 concludes the paper.

2. Related work

State of the art marker-less human pose estimation and tracking algorithms tend to fall into two categories. Discriminative approaches use large training datasets and machine learning techniques in order to map the extracted features from the input data to body parts and poses. Generative approaches, on the other hand, try to match the input data to articulated body templates by minimizing an objective function, utilizing various optimization techniques. There are also hybrid approaches which combine discriminative and generative techniques towards pose estimation and tracking. While initial implementations relied mainly on RGB data, the recent development of low-cost high-accuracy RGB-D sensors has pushed the research community towards approaches that utilize the sparse partial-view depth/3D data that these sensors offer.

Discriminative approaches, also known as single shot pose estimators, have been successfully used for human pose estimation, utilizing both RGB and Depth images [22]. They rely on large datasets and machine learning techniques in order to directly train the conditional probability of a body part within an image, thus providing robust human pose estimation from a single frame without requiring any prior knowledge regarding the human's position within the scene. The main drawback of these approaches is the requirement of extremely large and diverse datasets in order to generate the recognition models, which can be hard to acquire and train [33]. However, this process needs to be performed only once.

Towards human pose estimation from RGB images Bourdev and Malik [4] introduce the concept of "poselets", by extracting HOG [8] features, for human body part detection and Wang *et al*. [38] further extend it by introducing a structured hierarchical model for each poselet. Andriluka *et al*. [2] utilize a Pictorial Structure Model [13] for human pose estimation, while in [36, 7, 6] Convolutional Neural Network-based [20] methods for human pose estimation are proposed, taking advantage of available large pretrained networks which can be fine-tuned towards RGB-based body part estimation.

Utilizing Depth/3D information, Plagemann *et al*. [28] propose a novel interest point detector, called "Accumulative Geodesic EXtrema", which iteratively selects points of interest by incrementally maximizing geodesic distances on the surface of the 3D mesh, using Dijkstras algorithm [10]. Shotton *et al*. [33] use randomized decision trees and forests for body part detection and treat the body part segmentation as a per-pixel classification task. Similarly, Pons-Moll *et al*. [30] also utilize randomized decision forests, but propose an alternative training approach by employing the "Metric Space Information Gain" (MSIG) training objective. Targeting high performance on low-powered hardware, Jung *et al*. [18] also employ randomized decision trees for pose estimation. They achieve a large computation gain by substituting pixel-wise classification with the estimation of the probability distribution to the direction towards a particular joint. Meanwhile, in [24, 37] large synthetic depth human motion datasets are introduced, leading to CNN-based body part estimators from depth data.

Generative approaches, also known as body pose trackers, attempt to track the human pose by fitting an articulated human body template to the observed data. The pose of the body template is described by a pose vector and the fitting process involves the estimation of the optimal values of the vector that minimize an objective function, which expresses the similarity between the input data and the human template for a given pose. While they do not require any prior training, generative approaches do need a rough initial pose of the body, so that the optimization algorithm will be able to converge to the actual pose during the initialization step.

Ganapathi *et al*. [16] use a Dynamic Bayesian Network to model human body motion, which is modeled as a collection of linked 3D capsules, and introduce an enhanced ICP-based model [9] that utilizes free space constraints, termed "Ray-Constrained ICP Model", by applying Chamfer distance transforms. A generalization of Signed Distance Functions (SDF) [12] for articulated objects is introduced by Schmidt *et al*. [32]. Objects are represented by a symmetric version of the articulated SDF in 3D space, which can be precomputed, allowing for a dense representation of the data and gradient based optimization is employed to estimate the optimal pose, taking into consideration free space constraints similar to [16]. Ye and Yang [41] relate the observed data to a realistic skinned body template by assuming that the observed point cloud follows a Gaussian Mixture Model (GMM), while simultaneously performing shape adaptation by optimizing the human template in regard to limb length and body part geometry. The pose estimation and shape adaptation problem is then solved iteratively using the Expectation-Maximization (EM) algorithm. Towards achieving high performance for non-GPU optimized implementations, a Generalized Sum-of-Gaussians (G-SOG) model for human shape modeling is presented in [11]. The observed data are represented by isotropic Gaussians through octree partitioning, and a multivariate Gaussian kernel correlation-based objective function is optimized by employing a Quasi-Newton algorithm.

Hybrid approaches have also been proposed, combining single-shot estimators with articulated trackers. In these methods, pose tracking is usually performed through generative techniques, while discriminative algorithms work complementary to the main tracker proposing potential joint positions for faster convergence, initializing the pose tracker or recovering it from failure, leading to an overall improvement in robustness and accuracy, increasing, however, the implementation complexity as well.

Wei *et al*. [39] use a gradient-based optimizer to track the human pose, and combine it with a randomized decision trees-based body part detector similar to [33], used only for initialization and tracking failure recovery. In a similar fashion, Ganapathi *et al*. [15] also utilize a gradient-based optimizer, modeling the system as a DBN, similar to [16], in conjunction with the body part detector from [28]. However, in contrast to [39], the body part proposals are taken into consideration in every frame and not only during the initialization/reset phase. In [34], the authors further extend the random forests-based approach by learning to predict direct correspondences between input image pixels and a 3D articulated mesh model, with the optimization process being handled by a Quasi-Newton algorithm. Baak *et al*. [3] provide a rough pose estimation by selecting the pose most similar to the observation, from a database of pre-rendered mesh models, using the feature extraction algorithm from [28]. The final human pose is extracted by fusing the two alternative pose hypotheses, from the generative and the discriminative components, through a novel voting scheme based on sparse Hausdorff distance. Database lookup is also employed in [40], where PCA of normalized depth images is used to find an initial rough pose, which is then further refined through non-rigid registration between the observed point cloud and the estimated human pose.

Despite the significant progress in the domain of human pose estimation and tracking, there are still challenges that need to be overcome in order to achieve robust performance in real-life applications, especially in the scope of robotic applications for assisted living, where constraints in the available camera FOV due to the robot's positioning can severely hinder the accuracy of the body part estimation and tracking techniques [26, 29]. Towards this end, the main contributions of this paper are: (1) a real time hybrid human pose estimation and tracking framework with incorporated body part visibility and intersection models, (2) a tracking initialization and failure detection module, (3) a large object handling technique, (4) a custom human motion dataset captured under realistic conditions. The proposed framework is also comparatively evaluated in three public human motions datasets, achieving state-of-the-art performance, while also presenting promising results in unconstrained environment settings.

3. Articulated human pose tracker

The framework's human pose tracker employs the articulated SDF model [32] and uses an articulated skinned human template to track the human pose in sequences of depth images, extracting the 3D positions of the skeletal joints.

3.1. Articulated Human Template

The articulated human template used in our approach is created using the MakeHuman open source tool [35] and includes a skeleton composed of 13 rigid body parts connected to each other in a kinematic tree through 10 joints, along with 13000 vertices, each one rigidly attached to a single skeleton body part, which represent the shape of the human body.

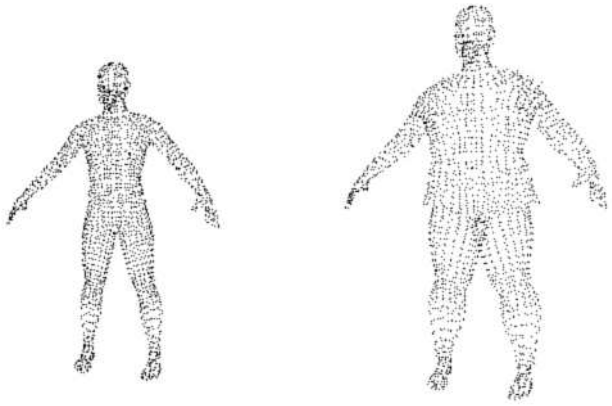

Figure 1. Human template customization. Left: [g=1,s=1], Right [g=1.2, s=1.5]

The template pose is described through a pose vector $\boldsymbol{\theta} = [\boldsymbol{t}_0, \boldsymbol{q}_0, \boldsymbol{q}_1, ..., \boldsymbol{q}_{10}]$ which includes the rotation of each body part $\boldsymbol{q}_{1...10}$ relative to its parent, along with the global rotation $\boldsymbol{q}_0$ and translation $\boldsymbol{t}_0$ of the template with reference to the camera frame. For efficiency reasons and to avoid the problem of gimbal lock [1], the relative rotations of each body part are represented in the pose vector using unit quaternions. Given a pose $\boldsymbol{\theta}_0$, the transformation matrix $T_{i,0}$ of a body part i in relation to the camera frame can be defined recursively by composing the transforms in a chain:

$$T_{i,0}\left(\boldsymbol{\theta}_0\right) = T_{prt(i),0}\left(\boldsymbol{\theta}_0\right) T_i\left(\boldsymbol{\theta}_0\right) \tag{1}$$

where T_i the transormation matrix of body part i in relation to its parent $prt(i)$.

Two shape factors are also introduced to allow the dynamic manipulation of the shape of the template: a global scale factor g which uniformly resizes the model, to account for variations in height and limb length, and a width factor s which alters the position of the skin vertices in order to fit on silhouettes of humans with variable body fat (Figure 1).

3.2. SDF-based tracker

The SDF-based human pose tracker attempts to minimize the sum of the distances between the 3D points of an observed point cloud and the 3D points of the mesh of the human template. In traditional ICP algorithms, this requires the explicit nearest point computation for each of the input cloud points. However, it is possible to remove the need for this taxing process by using implicit surface representations in the form of precomputed signed distance functions [14], which replaces nearest point computation with a much

faster lookup in the SDF. This approach is further extended in [32] with the introduction of an articulated model signed distance function $SDF_{mod}(\boldsymbol{x}, \boldsymbol{\theta}) : R^3 \rightarrow R$, which defines for all $\boldsymbol{x} \in R^3$ the distance to the surface of a model when articulated according to the pose vector $\boldsymbol{\theta}$. Thus the pose estimation process is achieved by estimating the pose vector $\hat{\boldsymbol{\theta}}$ that minimizes the tracker energy function:

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} SDF_{model}(\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\boldsymbol{x} \in \Omega} SDF_{mod}(\boldsymbol{x}, \boldsymbol{\theta}) \end{aligned} \tag{2}$$

where Ω the input 3D point cloud.

3.2.1 Model representation

For each rigid body part i of the human template, the 3D mesh points rigged to it form a geometry defined implicitly by a signed distance function $SDF^i(\boldsymbol{x}, \boldsymbol{\theta}) : R^3 \rightarrow R$ [12], which takes on negative values inside the geometry, positive values outside, and has a zero value at the surface interface. This allows to approximate the global signed distance function $SDF_{mod}(\boldsymbol{x}, \boldsymbol{\theta})$, as a composition of the precomputed local signed distance functions $SDF^i(\boldsymbol{x}, \boldsymbol{\theta})$, which alleviates the need to re-compute the full global SDF every time the pose is updated.

3.2.2 Data Association

In order to calculate the value of the energy function $SDF_{model}(\boldsymbol{\theta})$ for a given pose $\boldsymbol{\theta}_0$ the following steps are followed, similarly to [32]:

1. *Transformation of the observed point cloud to body part space*: All the points of the input point cloud are transformed to the local coordinates system of each rigid body part
2. *SDF lookup*: For each point of the input point cloud, the distance of the closest point on the surface of each body part is estimated by trilinear interpolation of the precomputed SDF of each body part
3. *Body part assignment*: Based on the distances estimated in the previous step, each point of the input cloud is assigned to a body part. If the distance is larger than d_{thresh}, the point is discarded as an outlier. The sum of the distances of each point to its corresponding body part is the value of the energy function.

3.2.3 Optimization

For the estimation of the optimal pose, a Quasi-Newton method is employed, using the BFGS algorithm [25], which in contrast to the Gauss-Newton algorithm used in [32], can minimize any general real-valued function $f(x)$ instead of only nonlinear least-squares problems.

A Quasi-Newton optimization method uses an iterative approach to arrive at a minimal function value, similar to gradient descent or Newton's method for optimization, requiring the Hessian of the function in order to estimate the direction of the line-search algorithm. Instead of explicitly supplying the Hessian, Quasi-Newton methods approximate the matrix using rank-one updates specified by gradient evaluations, which are simpler to evaluate. The gradient values are calculated by estimating the data association error for each point, as described in Subsection 3.2.2, and explicitly computing the first derivatives according to the kinematic model structure and current pose estimate. The resulting pose step $\boldsymbol{\Delta\theta}$ then updates the current pose estimate, and the algorithm iterates as needed until it meets a convergence criterion ($|\boldsymbol{\Delta\theta}| \approx 0$), which means that the optimization process has converged to a final pose vector.

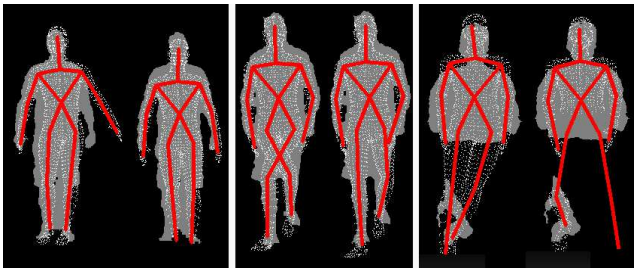

Figure 2. Pose correction through the complementary tracking features. Left: free space violation, Center: leg intersection, Right: body part visibility in occluded view; instead of converging to the visible right leg, the left occluded leg remains stationary.

4. Complementary tracking features

A series of complementary tracking features are introduced herein, in order to increase the overall robustness and accuracy of the framework's main tracking algorithm, described in Section 3, when operating under real life monitoring conditions (Figure 2).

4.1. Free space violation

Free space violation occurs when a part of the human template does not correspond to any of the input data and ends up "floating" in free space (Figure 2). To avoid this issue a free space constraint is introduced. Specifically, the 2D-SDF of the input depth image is computed [12]. Next, the deformed template is projected on the image, with each vertex of the template contributing towards a free-space error $SDF_{fs}(\boldsymbol{\theta})$, based on the value of the corresponding pixel on the 2D-SDF image. As a result, the energy minimization function (2) takes the form:

$$SDF_{overall}(\boldsymbol{\theta}) = SDF_{model}(\boldsymbol{\theta}) + \lambda SDF_{fs}(\boldsymbol{\theta})$$
$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} SDF_{overall}(\boldsymbol{\theta}) \quad (3)$$

where $\lambda \leq 1$ a weighing factor, which while not affecting the detection accuracy, helps the optimization algorithm converge faster to a pose, reducing the number of iterations.

4.2. Body part visibility

Commonly encountered in realistic environments is the occlusion of body parts due to obstacles and constraints of the camera's FOV. In order to ensure that in such cases only the visible body parts are used in the optimization process, a preprocessing step is employed, which calculates the visibility of each body part before moving to the optimization step. Specifically, the last tracked skeleton pose is projected on the camera's image plane and any body parts that are outside the camera's FOV or do not have valid depth observations in their surroundings pixels, are considered non-visible. For body parts without any children (lower arm, shin, head), visibility is determined by the position of that part's endpoint, while for the rest of the body parts the mid-point is used. Non-visible body parts are not taken into consideration during the optimization process, by setting to zero the contribution of their corresponding 3D points to the energy function $SDF_{overall}(\boldsymbol{\theta})$:

$$SDF^{i}(\boldsymbol{x}, \boldsymbol{\theta}) = 0 \quad \boldsymbol{x} \in \Omega_i, \quad \boldsymbol{\Delta q}_i = 0 \quad (4)$$

where i the non-visible body parts, Ω_i the input points that correspond to part i and $\boldsymbol{q}_i$ the quaternion that describes its relative rotation in the pose vector $\boldsymbol{\theta}$.

4.3. Leg intersection

Mix-up of the lower limbs can sometimes be observed (Figure 2), especially in cases of noisy observations between the legs (due to long distance from the camera or baggy clothes) or quick turn-arounds of the human which may create a local minimum that "traps" the optimization algorithm.

To counteract this issue, a leg intersection fix is employed. Using a body part representation similar to [34], each of the four lower body parts of the human template (hip R/L and shin R/L), is approximated by 7 spheres s (center $\boldsymbol{c}_s$, radius r_s) equally spaced along the body part. For a given pose $\boldsymbol{\theta}$, intersection between spheres s, t occurs when $|\boldsymbol{c}_s(\boldsymbol{\theta}) - \boldsymbol{c}_t(\boldsymbol{\theta})| < r_s + r_t$, thus we introduce the leg intersection error defined as:

$$E_{intr}(\boldsymbol{\theta}) = \sum_{(s,t)\in P} \frac{1}{1 + e^{-(r_s + r_t - |\boldsymbol{c}_s(\boldsymbol{\theta}) - \boldsymbol{c}_t(\boldsymbol{\theta})|)\gamma}} \quad (5)$$

where P is the set of pairs of spheres and γ a normalization factor.

In contrast to [34], where the intersection error is incorporated within the optimization process, we opt to execute the leg intersection correction at post-processing, taking advantage of the fact that only two subcases need to be taken into consideration (mix up between R/L hip or R/L shin). Specifically, if the error for the current pose is found larger than a threshold E_I, it is recalculated for two more poses: a) R/L knees interchanged and b) R/L ankles interchanged, with the pose that produces the minimum error retained as the optimal pose. Moving the intersection fix at post-processing ensures that any potential leg intersections will be fixed and removes the need to recalculate the error in each optimization iteration.

5. Framework initialization

The use of the BFGS algorithm, as with any iterative optimization technique, presents the risk of the optimization process converging to a minimal function value that does not correspond to the actual human pose, due to the fact that the objective function is not convex and can present local minima. As a result, it becomes essential that the tracking process begins from a pose close enough to the actual pose of the human.

To this end, a two-step initialization process is introduced herein, using as input the human pose estimation provided by the Kinect v1 built-in skeleton tracker [33] (Figure 3), in order to initialize the human template in the first frame of a tracking sequence. First the scale, global rotation and translation of the human template related to the camera frame are calculated, by estimating the rigid transformation between the torso area of the template (formed by the hips and shoulders joints) and the corresponding area of the observed human, taken from the Kinect v1 skeleton tracker. Next, each of the human template's body parts are initialized. Similarly to the main optimization process of the pose tracker (Sub-section 3.2.3), the initial pose $\boldsymbol{\theta}_{init}$ is estimated by minimizing, using the BFGS solver, the initialization energy function which is defined as the sum of the distances between the corresponding joints of the template and the target skeleton provided by the Kinect v1 skeleton tracker:

$$\boldsymbol{\theta}_{init} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i} |\boldsymbol{jt}_i(\boldsymbol{\theta}) - \boldsymbol{jt}_{i,init}| \quad (6)$$

where $\boldsymbol{jt}_i(\boldsymbol{\theta})$ and $\boldsymbol{jt}_{i,init}$ are the 3D positions of the joint i of the template and the initialization skeleton respectively.

5.1. Tracking confidence and re-initialization

After the conclusion of the initialization process, the pose tracker functions independently, using as an initial pose the pose estimated in the last frame. In order to ensure, however, that the last tracked pose is indeed correct

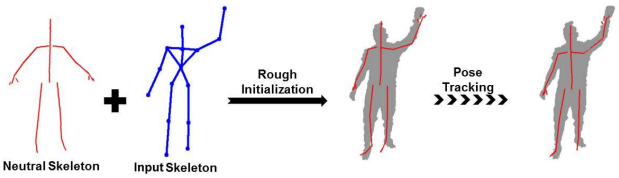

Figure 3. The initialization pipeline of the human pose tracker

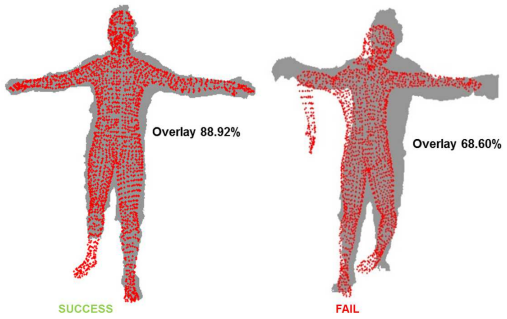

Figure 4. High (left) and low (right) confidence pose tracking. Deformed human template in red, input human silhouette in grey

and the optimization process did not converge to a wrong human pose, a metric is introduced to monitor the quality of the tracking process. The tracking confidence is estimated by projecting the deformed human template on the camera's image plane, taking into consideration only the visible body parts, and calculating its overlay ratio in respect to the input image. If the confidence score falls below a threshold, the estimated pose is considered incorrect (Figure 4) and the tracker is re-initialized.

6. Large object handling

Handling of objects by the monitored human is a very common scenario in realistic environments. In the case of small handheld objects, such as a cup, it does not significantly impact the accuracy of the pose tracking algorithm. However, when large objects, such as a cupboard door, are incorporated inside the foreground data that is passed to the algorithm, the optimization process can lead to erroneous results, as the optimizer tries to match the human template to the input human and the large object simultaneously, thus making it necessary to remove any non-human data from the input. Such cases are of particular importance for service robots that monitor human activities (e.g. cooking) in the context of assistive applications.

Towards this end, a data preprocessing step is introduced. Specifically, the last successfully tracked human silhouette, along with a small buffer zone, is projected on the new input depth image. Any large areas that do not overlap with the human silhouette are considered as candidate large objects. Taking advantage of the fact that large objects tend to be smooth surfaces (i.e. doors, tables etc),

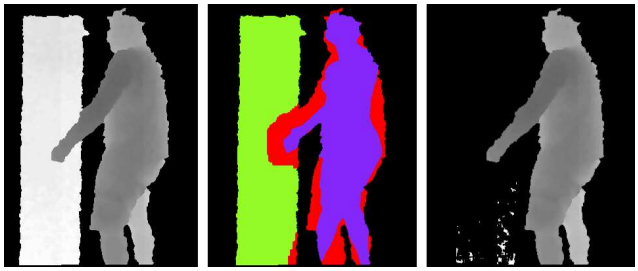

Figure 5. Data preprocessing for large object removal. The input image is split in three regions: blue - overlap with the last tracked human silhouette, red - buffer zone, green - candidate large object area which provides the seeds for the flood-fill algorithm. The image on the right is the processed image passed to the optimizer

the flood-fill algorithm is used on the depth image, seeded from the center point of each candidate area, in order to remove any large objects in the scene. The threshold value for the flood-fill algorithm is set to a relatively low value to ensure that no parts of the actual human will be removed, resulting in small parts of the object being left in the scene as well, without, however, significantly impacting the algorithm's accuracy anymore. Figure 5 presents an example of the large object removal process.

7. Experimental evaluation

Following the evaluation process employed in the relevant literature, the metrics used for the experimental evaluation of the proposed framework are: a) *Average Distance Error (ADE)*: Average distance between the ground truth and the estimated joint positions, and b) *Mean Average Precision at 0.1m (mAP)*: Percentage of body joints where the distance between the ground truth and the estimated joint position is less than $0.1m$. In all the experimental sequences a generic human model is used, which is initialized in the first frame using the ground truth data, while the proposed framework performs at an average framerate of 24fps on a modern CUDA-enabled GPU. Detailed experimental results can be found in the supplemental material.

7.1. Individual components evaluation

Two small-scale experiments are conducted, using subsets of the publicly available PDT [17] dataset, in order to evaluate the effectiveness of the body part visibility and leg intersection features.

The *body part visibility* component is tested in sequences of moderate lower body motion complexity, with the lower body parts being artificially occluded, by removing all the input data below the subjects' hips. Enabling the visibility component does not affect significantly the upper body parts' accuracy (disabled: [*ADE* 0.030, *mAP* 0.983] - enabled: [*ADE* 0.032, *mAP* 0.974]), which is expectable as all the upper body parts are visible throughout the testing se-

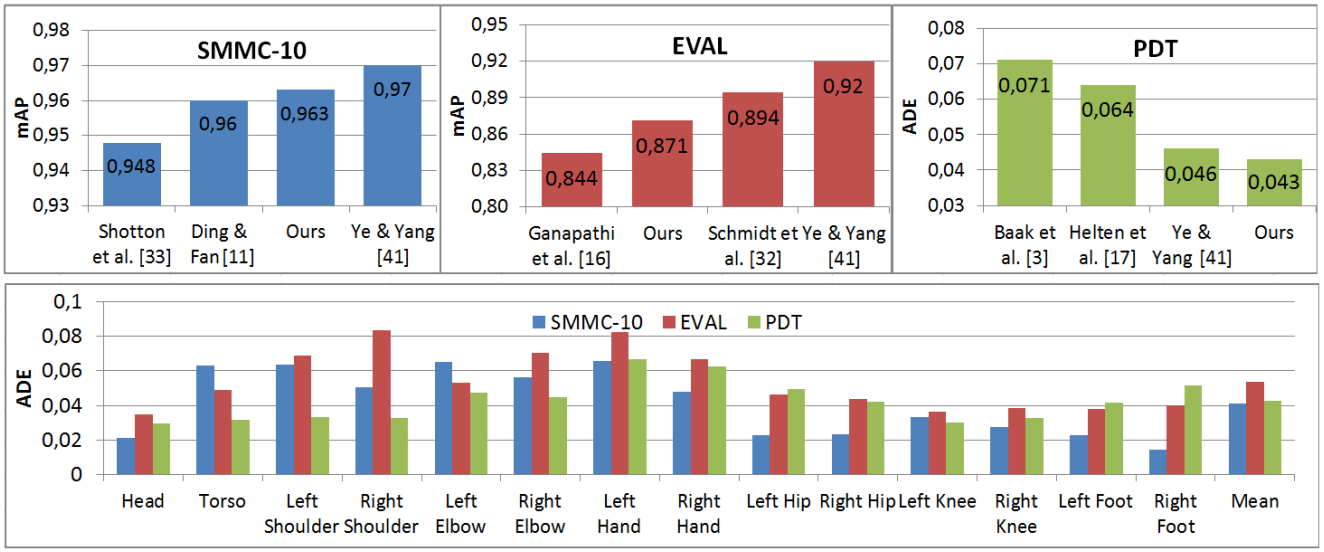

Figure 6. Framework performance on three public datasets. Top - comparison to state-of-the-art methods (results as reported by their authors), Bottom - *ADE* per-joint

quences. For the occluded lower body parts, on the other hand, there is a clear increase in accuracy when enabling the body part visibility component (disabled: [*ADE* 0.283, *mAP* 0.366] - enabled: [*ADE* 0.204, *mAP* 0.389]). While the improvement in *mAP* is not significant, which is to be expected since it is hard to reach the 0.1m threshold without any information about the body parts, there is major improvement in the *ADE* of almost $8cm$, which indicates that the body part visibility component keeps the occluded body parts closer to their actual position, increasing the probability of the tracker correctly tracking the occluded body parts once they reappear.

The *leg intersection* feature is tested in sequences with relatively complex leg movement. With the intersection feature disabled, the tracker sometimes mixes the lower body parts and keeps them mixed throughout the sequence, resulting in low accuracy for the four leg joints (R/L knee, R/L foot): *ADE* 0.094, *mAP* 0.814. However, enabling the leg intersection feature limits the leg mix-up to a few frames, significantly increasing the lower body accuracy to: *ADE* 0.041, *mAP* 0.941.

7.2. Public datasets

The overall developed pose tracking method is evaluated using three publicly available human pose tracking datasets: SMMC-10 [15], EVAL [16] and PDT [17]. All three datasets were recorded using a TOF/depth camera, and include single subjects performing sequences of variable motion complexity and human pose ground truth data provided from a marker-based motion capture system.

Figure 6 presents the performance comparison between our framework and current state-of-the-art methods (as reported by their authors), along with the per-joint accuracy for each of the three datasets. From the experimental results it becomes evident that the proposed framework performs comparably to other state-of-the-art methods, approaching the accuracy of the top performers in the SMMC-10 and EVAL datasets, while slightly outperforming them in the PDT dataset.

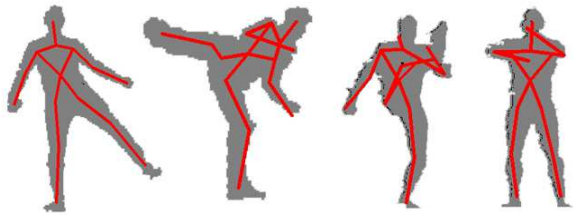

Figure 7. Sample frames from the evaluation on public datasets.

7.3. Realistic human motion dataset

In order to evaluate the overall performance and robustness of the proposed framework in real-life monitoring conditions, a realistic human-motion dataset[1] is captured, since the public datasets used in subsection 7.2 were recorded in controlled laboratory environments and lack in challenges encountered in unrestrained real-life settings.

The dataset is captured using a Kinect v1 depth camera positioned on-board of a service robot at height $h \approx 1.2m$, simulating scenarios where the subject's domestic activities are monitored by an assistive service robot. In total 11 subjects (10 male, 1 female) perform a series of everyday actions in a single 1.5-minute long sequence including: inter-

[1]available online: http://www.ramcip-project.eu/ramcip-data-mng

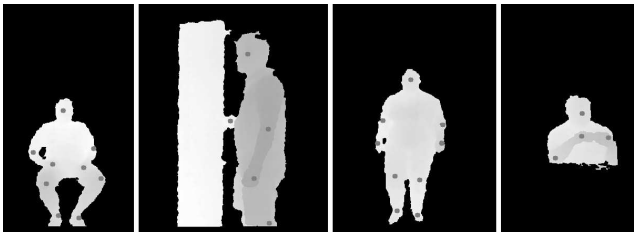

Figure 8. Sample frames from the realistic human motion dataset, with manually annotated ground truth data.

action with large (i.e cupboard door) and small (i.e. pillbox) objects, sitting on a chair, eating while occluded behind a table, moving around the room near the FOV limits or extremely close/far from the camera etc. This resulting human motion dataset incorporates, in contrast to currently available public datasets, many of the challenges encountered in real life scenarios, such as: occlusions due to the presence of obstacles, out-of-FOV clipping, noisy observations due to object handling, decreased sensor accuracy due to long camera/subject distance, long-duration side-view tracking, non-frontal complex initialization poses, large variety in human body shape and more, creating a realistic benchmark for the evaluation of human pose estimation techniques in real-life settings.

Ground truth annotation is performed manually on the recorded data, by selecting and 3D projecting 9 skeleton joints: *head, R/L elbow, R/L hand, R/L knee, R/L ankle*. The selection of the joints is based on ensuring consistency among consecutive frames (i.e. a *torso* joint would be hard to track consistently), while annotation is performed every 10 frames and includes only joints visible in the specific frame. Figure 8 shows some sample frames from the realistic human motion dataset.

The proposed framework is tested on the realistic human motion dataset, achieving promising results: *ADE* 0.075, *mAP* 0.825, and clearly outperforming the Kinect built-in pose estimator (NITE 2.2 middleware [31][2]). Figure 9 presents the detailed results, while Figure 10 shows some sample frames from the evaluation process.

8. Future work and conclusions

This paper introduces a human pose estimation and tracking framework capable of accurate and robust real time performance in real-life applications, targeting mainly the domain of robotic applications for assisted living. The framework builds upon the articulated-SDF tracking approach and further enhances its robustness by introducing a series of complementary mechanisms to tackle problems that arise under real-life conditions, such as tracker initialization and failure, pose tracking from partial-views, large

[2]while the NITE algorithm has not been published, its functionality and performance approach the ones of Shotton *et al*. [33, 5]

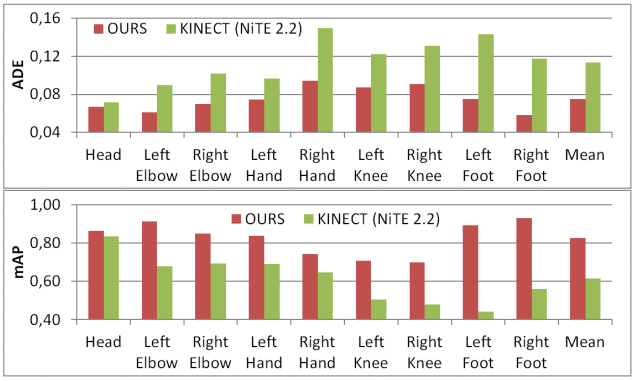

Figure 9. Comparison of per-joint *ADE* and *mAP* with the Kinect built-in pose estimator (NITE 2.2 middleware [31]) on the realistic human motion dataset.

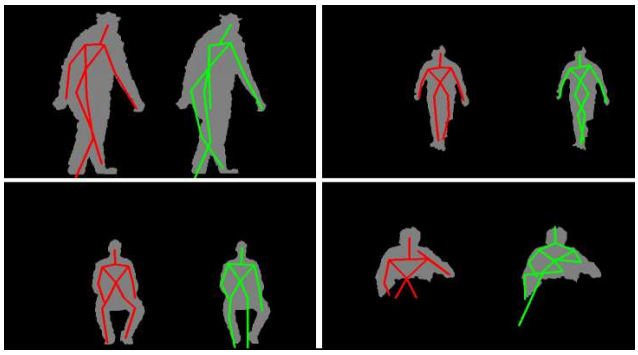

Figure 10. Sample frames from the evaluation on the realistic human motion dataset: Red - proposed framework, Green - Kinect built-in pose estimator (NITE 2.2 middleware [31])

object handling and body part intersection. Through experimental evaluation on a new realistic human motion dataset, specifically targeting these problems, the proposed framework is found to outperform currently available pose estimation techniques.

Future improvements of the proposed framework may include incorporation of the discriminative estimator within the optimization process and enhancement of the accuracy utilizing CNNs [24, 37], which have achieved impressive results in identification tasks [20]. Further elaboration of the data preprocessing algorithms can also increase robustness, while from a h/w standpoint, performance modelling on various architectures may potentially provide significant benefits, as achieving robust and accurate pose estimation on lower-spec h/w can be of major significance in the case of applications where the available processing power is limited, such as autonomous robots.

Acknowledgements

This work has been supported by the EU Horizon 2020 funded project "Robotic Assistant for MCI Patients at home (RAMCIP)" under the grant agreement no. 643433.

References

[1] W. Alan and W. Mark. Advanced animation and rendering techniques. *Addison-Wesley*, 1992.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021. IEEE, 2009.

[3] 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*, pages 71–98. Springer, 2013.

[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1365–1372. IEEE, 2009.

[5] L. V. Calderita, J. P. Bandera, P. Bustos, and A. Skiadopoulos. Model-based reinforcement of kinect depth data for human motion capture applications. *Sensors*, 13(7):8835–8855, 2013.

[6] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[7] C.-H. Chen and D. Ramanan. 3d human pose estimation= 2d pose estimation+ matching. *arXiv preprint arXiv:1612.06524*, 2016.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[9] D. Demirdjian, T. Ko, and T. Darrell. Constraining human body tracking. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1071–1078 vol.2, Oct 2003.

[10] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[11] M. Ding and G. Fan. Articulated gaussian kernel correlation for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 57–64. IEEE, 2015.

[12] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell University, 2004.

[13] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[14] A. W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13):1145–1153, 2003.

[15] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 755–762. IEEE, 2010.

[16] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. *Real-Time Human Pose Tracking from Range Data*, pages 738–751. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[17] T. Helten, A. Baak, G. Bharaj, M. Muller, H.-P. Seidel, and C. Theobalt. Personalization and evaluation of a real-time depth-based full body tracker. In *Proceedings of the 2013 International Conference on 3D Vision*, 3DV '13, pages 279–286, Washington, DC, USA, 2013. IEEE Computer Society.

[18] H. Y. Jung, S. Lee, Y. S. Heo, and I. D. Yun. Random tree walk toward instantaneous 3d human pose estimation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2467–2474, June 2015.

[19] I. Kostavelis, D. Giakoumis, S. Malasiotis, and D. Tzovaras. Ramcip: Towards a robotic assistant to support elderly with mild cognitive impairments at home. In *International Symposium on Pervasive Computing Paradigms for Mental Health*, pages 186–195. Springer, 2015.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[21] M. Leo, G. Medioni, M. Trivedi, T. Kanade, and G. M. Farinella. Computer vision for assistive technologies. *Computer Vision and Image Understanding*, 154:1–15, 2017.

[22] Z. Liu, J. Zhu, J. Bu, and C. Chen. A survey of human pose estimation: the body parts parsing based methods. *Journal of Visual Communication and Image Representation*, 32:10–19, 2015.

[23] T. B. Moeslund, A. Hilton, and V. Krger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(23):90 – 126, 2006. Special Issue on Modeling People: Vision-based understanding of a persons shape, appearance, movement and behaviour.

[24] K. Nishi and J. Miura. Generation of human depth images with body part labels for complex human pose recognition. *Pattern Recognition*, 2017.

[25] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[26] S. Obdrzalek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel. Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1188–1193, Aug 2012.

[27] S. I. Park and J. K. Hodgins. Capturing and animating skin deformation in human motion. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 881–889. ACM, 2006.

[28] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3108–3113. IEEE, 2010.

[29] P. Plantard, E. Auvinet, A.-S. L. Pierres, and F. Multon. Pose estimation with a kinect for ergonomic studies: Evaluation of the accuracy using a virtual mannequin. *Sensors*, 15(1):1785–1803, 2015.

[30] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon. Metric regression forests for correspondence estimation. *International Journal of Computer Vision*, 113(3):163–175, 2015.

[31] PrimeSense. NiTE: Natural Interaction Middleware. `http://openni.ru/files/nite`.

[32] T. Schmidt, R. Newcombe, and D. Fox. Dart: dense articulated real-time tracking with consumer depth cameras. *Autonomous Robots*, pages 1–20, 2015.

[33] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[34] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 103–110. IEEE, 2012.

[35] The MakeHuman team. MakeHuman: Open source tool for making 3D characters. `http://www.makehuman.org`.

[36] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660. IEEE, 2014.

[37] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from Synthetic Humans. In *CVPR*, 2017.

[38] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *2009 IEEE 12th International Conference on Computer Vision*, pages 32–39. IEEE, 2009.

[39] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Transactions on Graphics (TOG)*, 31(6):188, 2012.

[40] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3d pose estimation from a single depth image. In *2011 International Conference on Computer Vision*, pages 731–738. IEEE, 2011.

[41] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2345–2352. IEEE, 2014.