sensors

MDPI

Article

Dual Quaternions as Constraints in 4D-DPM Models for Pose Estimation

Enrique Martinez-Berti *, Antonio-José Sánchez-Salmerón and Carlos Ricolfe-Viala

Departamento de Ingeniería de Sistemas y Automática, Instituto de Automática e informática Industrial, Universitat Politècnica de València, València, 46022, Spain ; asanchez@isa.upv.es (A.-J.S.-S.); cricolfe@isa.upv.es (C.R.-V.)

* Correspondence: enmarbe1@etsii.upv.es

Received: 1 June 2017; Accepted: 13 August 2017; Published: 19 August 2017

Abstract: The goal of this research work is to improve the accuracy of human pose estimation using the Deformation Part Model (DPM) without increasing computational complexity. First, the proposed method seeks to improve pose estimation accuracy by adding the depth channel to DPM, which was formerly defined based only on red–green–blue (RGB) channels, in order to obtain a four-dimensional DPM (4D-DPM). In addition, computational complexity can be controlled by reducing the number of joints by taking it into account in a reduced 4D-DPM. Finally, complete solutions are obtained by solving the omitted joints by using inverse kinematics models. In this context, the main goal of this paper is to analyze the effect on pose estimation timing cost when using dual quaternions to solve the inverse kinematics.

Keywords: DPM; 4D-DPM; dual quaternions; Kalman filter; polishphere; pose estimation; kinematic constraints

1. Introduction

Human pose estimation has been extensively studied for many years in computer vision. Many attempts have been made to improve human pose estimation with methods that work mainly with monocular red–green–blue (RGB) images such as [1–5].

With the ubiquity and increased use of depth sensors, methods that use red–green–blue-depth RGB-D imagery are fundamental. One of the methods that used such imagery, and which is currently considered the state of the art for human pose estimation, is Shotton et al. [6], which was commercially developed for the kinect device (Microsoft, Redmond, WA, USA). Shotton's method allows real-time joint detection for human pose estimation based solely on depth channel. Despite the state-of-the-art performance of [6] and the commercial success of kinect, the many drawbacks of [6] make it difficult to be adopted in any other type of three-dimensional (3D) computer vision system.

Some of the drawbacks of [6] include copyright and licensing issues, which restrict the use and implementation of the algorithm for working on any other devices. Another drawback of the algorithm is the large number of training examples (hundreds of thousands) that are required to train its deep random forest algorithm, and which could make training cumbersome. Another drawback of [6] is that its model is trained only on depth information, and thus discards potentially important information that could be found in the RGB channels and could help approach human poses more accurately. To alleviate these and other drawbacks in [6], we propose a novel approach that takes advantage of both RGB and depth information combined in a multi-channel mixture of parts for pose estimation in single frame images coupled with a skeleton constrained linear quadratic estimator (Kalman filter) that uses the rigid information of a human skeleton to improve joint tracking in consecutive frames. Unlike kinect, our approach makes our model easily trainable even for nonhuman poses. By adding depth information, we increase the time complexity of the proposed method. For this reason, to speed up the proposed method, we reduced the number of points modeled in the proposed method compared with the original deformation part model DPM. Finally, we propose an inverse kinematics method for the inference of the joints not considered initially, which cuts the training time.

The main contribution of our method extends to: (i) a multi-channel mixture of parts model that allows the detection of parts in RGBD images; (ii) a linear quadratic estimator (KF) that employs rigid information and connected joints of human pose; (iii) a model for unsolved joints through inverse kinematics that allows the model to be trained with fewer joints and in less time. In our previous work, [7,8], it is shown that computational cost is too high. This is the reason why in this paper a dual quaternion solution is introduced to improve the computational cost of the previously proposed method. Our results show significant improvements over the state of the art in both the publicly available CAD60 data set and our own data set.

Related Work

Human pose estimation has been intensely studied for decades in the field of computer vision due to its wide applications. Some of the methods in the literature that attempt to solve this problem date back to the use of pictorial structures (PS) introduced by [9]. More recent methods improve the concept of PS with improved features or inference models, as in [3,10–13]. Recently, the launch of low-cost RGB-D sensors (e.g., kinect) has further triggered a large amount of research due to their good performance from extra depth information whose intensities depict an inversely proportional relationship between the distance of the objects to the camera. The existing algorithms can be roughly categorized into three groups, i.e., using only RGB sensor, using only Depth sensors, or using both RGB and Depth sensors. Some approaches in the first group are [1,14–18]. Some approaches in the second group are [6,19–38]. Some approaches in the third group are [39,40].

Using RGB sensors, Yang et al. [1] uses a mixtures of parts model based on a robust joint relationships, Sapp et al. [14], in turn, uses a multimodal decomposable model, Bourdev et al. [16] addresses the classic problems of detection, segmentation and pose estimation of people in images with a novel definition of a part, a poselet, and Wang et al. [15] considers part-based models by introducing hierarchical poselets. Ionescu et al. [17] describes automatic 3D human pose reconstruction from monocular images, based on a discriminative formulation with latent segmentation inputs.

Using depth sensor, Shotton et al. [6], which was developed for the kinect algorithm, has become the state of the art for performing human pose estimation that predicts 3D positions of body joints from a single depth image. As mentioned in [26], to capture the human pose efficiently from multi-view video sequences, a sum of Gaussian (SoG) model was developed in [32]. This simple yet effective shape representation provides a differentiable model-to-image similarity function, allowing a fast and accurate full body pose estimation. The SoG model was also used in [33,36,40] for human or hand pose estimation. Extended from SoG, a generalized SoG model (GSoG) was proposed in [34], where it encapsulated fewer anisotropic Gaussians for human shape modeling, and a similarity function between GSoG and SoG was derived in 3D space. Meanwhile, a sum of anisotropic Gaussians (SAG) model [37] shared the similar spirit with GSoG for hand pose estimation, and it provided an overlap measurement between projected SAG and SoG/SAG in 2D image. Although GSoG and SAG based approaches have improved the pose estimation performance with better model adaptability, their similarity functions are specifically designed for different situations/applications. In addition, the clamping function that aims to handle the model intersection problem in previous SoG-based approaches [32,34,36] leads to a discontinuous energy function that could hinder the gradient-based optimization. In [26], inspired by the classical Kernel Correlation-based algorithm [38], generalizes previous SoG-based methods and derives a unified similarity function from the perspective of Gaussian kernel correlation. Ding et al. [26] embeds a kinematical skeleton into the kernel correlation, which enables us to achieve a fast articulated pose estimation.

Using both RGB and depth sensors, object detection has been done using RGB-D with Markov Random Fields (MRFs) and features from both RGB and Depth [39]. Ding et al. [40] defines a method that can capture a broad range of articulated hand motions at interactive rates.

The proposed method uses both RGB and Depth information and a discriminative method using a deformable parts model combined with a generative method using Kalman filter for tracking the human pose.

We first explain the proposed method, Section 2, using the pre-processing step, Section 2.1, for the depth channels in which the background was removed to improve the accuracy of our algorithm (see Figure 1). Section 2.2 explains the formulation of our four dimensional (4D) mixture of parts model. Section 2.3 explains our structured quadratic linear estimator for correcting joints in consecutive frames. Section 2.4 explains the polisphere model used. Finally, the Section 2.5 describes the strategy to reduce the computational complexity of our proposed method using dual quaternions. Finally, Section 3 shows us the results obtained comparing the proposed method (4D-DPM) with the original method DPM in Section 3.1 and a time complexity analysis in Section 3.2.

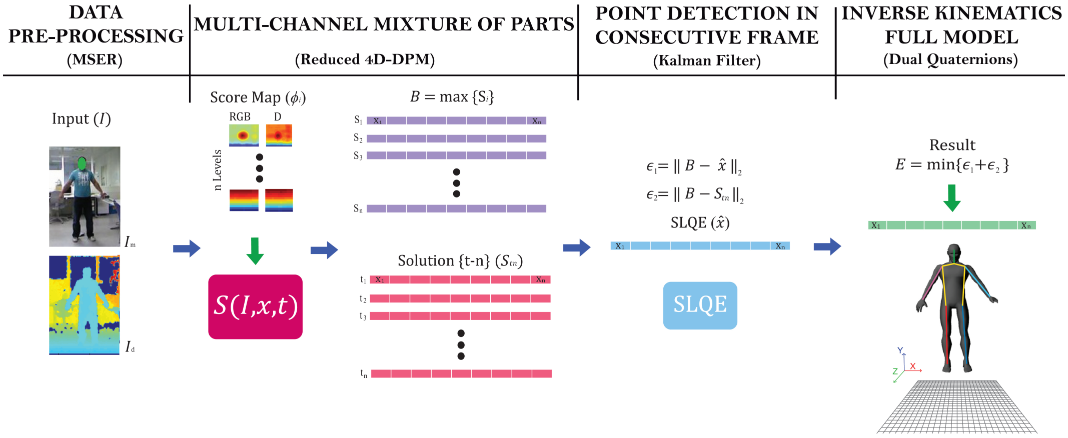


Figure 1. Outline of our method.

2. Proposed Method

2.1. Data Pre-Processing

As a processing step of RGB channels, we isolate significant foreground areas in these channels from background noise. This is done by removing regions in the depth images that are most unstable to different thresholds that belong to the background. Such a foreground and background template is then transferred to the RGB images to thus remove noise or conflicting object patterns that would confuse foreground and background features in our method, and would hinder detection accuracies.

The intuition behind this approach is that objects or people in the foreground seen through the depth sensor share areas with similar pixel intensities. The reason for this is that the infra red (IR) rays being reflected from the objects in the foreground are reflected more or less at the same time and with the same intensity. Other objects or areas that are much farther away from the IR camera unevenly reflect such rays, and these areas appear noisier and with varying intensities. Figure 2 shows the different intensities reflected from the IR sensor that represents the depth coordinates of the objects.

Due to this property of the pixel intensities in the depth images, our background removal method, which is used for depth and later applied to the RGB images, uses a maximally stable extremal regions (MSER) based approach [41]. These regions are the most stable ones within a range of all possible threshold values being applied to them. A stability score δ of each region in the depth channels is calculated so that $\delta = \frac{|\Delta R - R|}{|R|}$, where $|R|$ represents the area of the region in question and Δ represents the intensity variation for the different thresholds. Hence, we remove those MSER regions in which areas are above a T threshold. We train the parameters for MSER based on a subset of the training set.

We can see in Figure 2 the results from our background subtraction method. Note that most of the noisy pixels in the background have been removed.

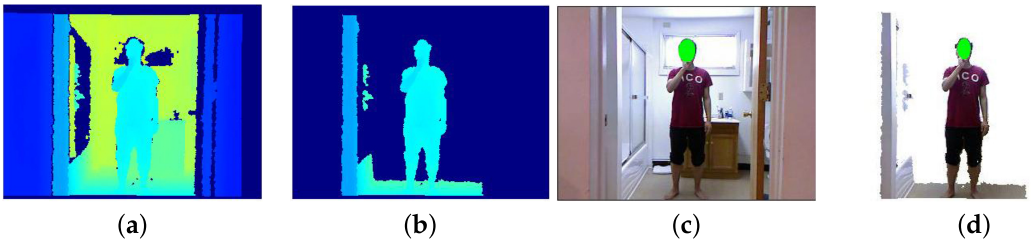


Figure 2. Pre-Processing: (**a**) original depth; (**b**) depth after applying maximally stable extremal regions (MSER); (**c**) original RGB; (**d**) combining image (**c**, **b**).

2.2. Multi-Channel Mixture of Parts

Until recently, Yang and Ramanan's method [1] has been a state-of-the-art method for pose estimation in monocular images. Yang and Ramanan's method performs poorly on images that vary from those in its training set, and their method only improves by a small margin even after retraining.

Although there have been other algorithms that have improved Yang and Ramanan's model, such as [2,3,5], all of these methods, including Yang and Ramanan's, use a mixture of parts for only the RGB dimension of channels. Conversely, our method uses a multi-channel mixture of parts model that allows us to extend the number of mixtures of parts to the depth dimension of RGBD images.

The depth channel increases time complexity, but this disadvantage has been solved by cutting the number of joints modeled in our 4D-DPM method. On [7,8,42], we can find the main equations changed to introduce the new dimension, depth channel.

2.3. Point Detection in Consecutive Frames

To date, we have dealt only with pose estimation for each single frame independently. However, most of the joint movement performed in normal circumstances displays uniform and constant changes of displacement and velocity. Hence, we can use joint velocity and acceleration to predict where joints would most likely be, given their past history. This motion-based prediction could help us validate our frame-based prediction.

One way of predicting joint location based on previous detections is by using a linear quadratic estimator (LQE). Using a simple LQE works well when the joints being tracked are independent of each other and their movement does not correlate. However, in our case, our joints are connected to each other through limbs, which are rigid connections and allow the movement of one joint related to the other one to be connected; e.g., the foot joint movement would be relative to a parent joint like as a knee or hip.

Using the same algorithm as [7,8,42], a Kalman filter is used for tracking the points of interest.

2.4. Geometric Model

In the case of improving the results of the Kalman filter, we introduce some restrictions using the geometric model. To do that, we use a polisphere to represent the human body. This representation allows us to detect collisions between the different parts of the body.

In Figure 3, we can see the geometric model used. Green parts are the principal spheres used and delimit each part of the body.

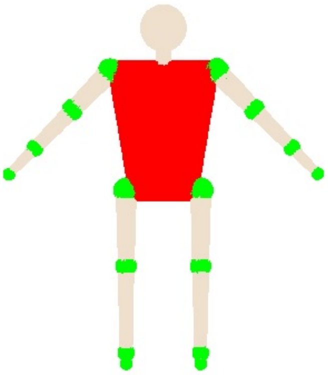


Figure 3. Geometric model using polispheres.

2.5. Model Simplification

The additional depth images included in our formulation add computational cost to our training and testing phases.

In this section, we explain a simplification technique that uses inverse kinematic equations in order to infer shoulder and knee joints. The original DPM model calculates the full body parts with 14 joints. By using inverse kinematics, we can lower that number of points to 10. The joints modeled in our proposed 4D-DPM method were reduced, as were the variables to be predicted with KF.

- **Human body model:** In order to track the human skeleton, we model it as a group of kinematic chains, where each part and joint in the human body corresponds to a link and joint in a kinematic chain. Given the joint positions predicted by the KF, inverse kinematics are used to obtain all of the joints using Dual Quaternions (DQ).
- **State variables:** The human body model is divided into four main kinematic chains (KC) that perform collision detection with their correspondent state variables, in essence: one KC for each arm and one for each leg. Figure 4 shows the state variable for each KC.

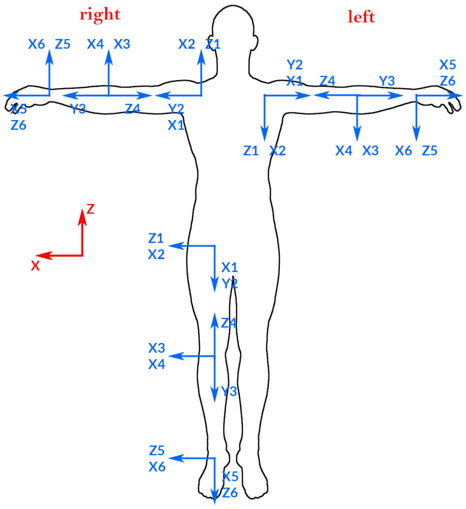


Figure 4. Coordinate systems used.

- **DQ model:** We use DQ to model each KC. In this sense, we use six joints for each KC for shoulders, hips, hands, and feet (see Figure 4).
- **DH model:** We use the Denavit-Hartenberg (DH) method to obtain the base coordinate system for each joint. After that, we will apply the dual quaternion method. First, we establish the base coordinate system (X_0, Y_0, Z_0) at the supporting base with the Z_0 axis lying along the axis

of motion of joint 1. We have four base coordinate systems (X_0, Y_0, Z_0), each one located at (X_1, Y_1, Z_1) from each KC. Then, we establish a joint axis and align the Z_i with the axis of motion of joint $i + 1$.

We also locate the origin of the i_{th} coordinate at the intersection of the Z_i and Z_{i-1} or at the intersection of a common normal between the Z_i and Z_{i-1}. Then, we establish $X_i = \pm (Z_{i-1} \times Z_i) \,/\, \|Z_{i-1} \times Z_i\|$ or along the common normal between the Z_i and Z_{i-1} axes when they are parallel. We also assign Y_i to complete the right-handed coordinate system. Finally, we find the link and joint parameters: θ_i (angle of the joint with respect to the new axis), d_i (offset of joint along the previous axis to the common normal), a_i (length of the common normal), and α_i (angle of the common normal with respect to the new axis).

For each KC, we have six variable joints q_i. Each q_i is placed on the z_i axis in Figure 4. (the left leg in Figure 4 has the same coordinate systems as the right leg.)

Once we have the coordinate systems for each joint, a dual quaternion method is explained. First, we introduce a DQ representation and then we explain the kinematics. A DQ is:

$$\hat{q} = (\hat{q}_s, \hat{q}_v) \qquad or \qquad \hat{q} = q + \varepsilon \hat{q}^0, \tag{1}$$

where $\hat{q}_s$ is a dual scalar, $\hat{q}_v$ is a dual vector, q and q^0 are two quaternions and ε is a dual unit. We define the next expressions:

$$\begin{array}{cc} q = (q_0, q_1, q_2, q_3) & \hat{q} = \begin{bmatrix} q_4 & q_5 & q_6 & q_7 \\ q_8 & q_9 & q_{10} & q_{11} \end{bmatrix} \\ \hat{q}_s = q_s + \varepsilon \hat{q}_s^0 & \hat{q}_v = q_v + \varepsilon \hat{q}_v^0 \\ V\{q\} = q_v = [q_1, q_2, q_3] & S\{q\} = q_s = q_0 \\ S\{R\{\hat{q}\}\} = q_s = q_4 & S\{D\{\hat{q}\}\} = q_s^0 = [q_5, q_6, q_7] \\ V\{R\{\hat{q}\}\} = q_v = q_8 & V\{D\{\hat{q}\}\} = q_v^0 = [q_9, q_{10}, q_{11}]. \end{array} \tag{2}$$

These equations represent different parts of quaternions product: $V\{q\}$ is a vectorial part, $S\{q\}$ is a scalar part, $S\{R\{\hat{q}\}\}$ is a scalar part of real part, $S\{D\{\hat{q}\}\}$ is a scalar part of dual part, $V\{R\{\hat{q}\}\}$ is a vectorial part of real part and $V\{D\{\hat{q}\}\}$ is a vectorial part of dual part.

All the movements of the rigid body in the 3D space, with the exception of pure translation, are equivalent to the screw movements, that is, the rotation on the line together with the translation on the line.

If the line passes over the origin, the movement of screw can be written as:

$$T = \begin{bmatrix} R(\theta, d) & \frac{\theta}{2\pi} pd \\ 0 & 1 \end{bmatrix}, \tag{3}$$

where $R(\theta, d)$ represents the 3×3 rotation matrix on the axis in the direction of the unit vector d through an angle θ.

If the axis of the screw movement does not pass over the origin, it can be written as:

$$T = \begin{bmatrix} I_{3x3} & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R(\theta, d) & \frac{\theta}{2\pi} pd \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_{3x3} & -p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R(\theta, d) & \frac{\theta}{2\pi} pd + (I_{3x3} - R(\theta, d))p \\ 0 & 1 \end{bmatrix}. \tag{4}$$

We can represent a screw movement as dual quaternions as follows:

$$\hat{q} = cos\left(\frac{\hat{\theta}}{2}\right) + sin\left(\frac{\hat{\theta}}{2}\right)\hat{d}, \tag{5}$$

where $\hat{\theta} = \theta + \epsilon k$ and $\hat{d} = d + \epsilon m$. θ is the rotation of screw angle, and $d = [0, d]$ is the movement of screw axis. The moment of the axis is $m = [0, p \times d]$. The point p is in the direction of d. And $k = d \cdot t$.

In a Plücker coordinates, each line can be fully represented by an ordered set of two vectors. The first point is a vector p that indicates the position of an arbitrary point on the line, and the second point is the direction vector d, which gives us the direction of the line. The Plücker coordinate can be represented as follows:

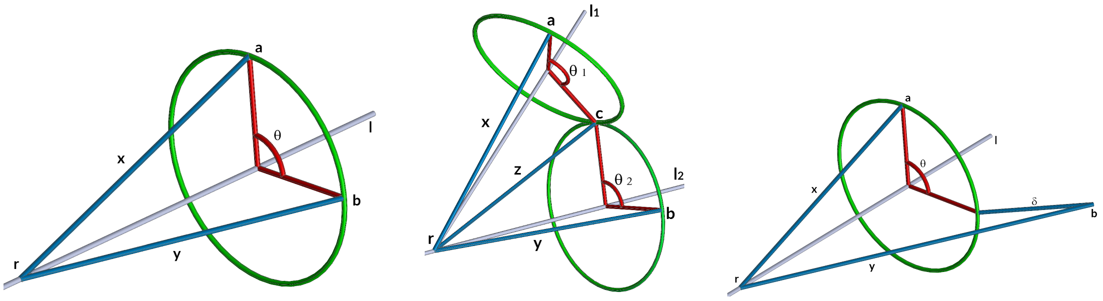
$$L_a(m, d), \tag{6}$$

where $m = p \times d$ is the vector moment of d with respect to the reference origin selected.

We can represent one line on Plücker coordiante as $\hat{l}_a = l_a + \epsilon m_a$, and we can transform that expression to $\hat{l}'_b = \hat{q} \odot \hat{l}_a \odot \hat{q}^*$ using the dual quaternion unit.

The representation of the Plücker coordinates is not minimal since it uses six parameters for the representation of the line. The main advantage of the representation of the Plücker coordinates is that it is homogeneous. $L_p(m, d)$ represents the same line as $L_p(km, kd)$, where $k \in \Re$.

To solve the forward and inverse kinematics using dual quaternions, we use Paden–Kahan subproblems. We have three sub-problems of Paden–Kahan, and Figure 5 shows graphically the three sub-problems.



(**a**) (**b**) (**c**)

Figure 5. Paden–Kahan sub-problems: (**a**) sub-problem 1; (**b**) sub-problem 2.; (**c**) sub-problem 3.

To solve the sub-problem 1, we have $y = q \otimes x \otimes q^*$ as the general movement equation, where $\theta = arctan2(S\{l \otimes x^{'} \otimes y^{'}\}, S\{x^{'} \otimes y^{'}\})$, $x' = x + S\{l \otimes x\}l$, and $y' = y + S\{l \otimes y\}l$, $l = [0, l]$ is the director vector of l.

To solve the sub-problem 2, we have $y = q_1 \otimes q_2 \otimes x \otimes q_2^* \otimes q_1^*$ as the general movement equation, $z = c - r$, $z = [0, z]$. If $l_1, l_2, l_1 x l_2$ are linearly independent, and we have $z = \alpha l_1 + \beta l_2 + \gamma[0, V\{l_1 \otimes l_2\}]$, where $\alpha = \frac{S\{l_1 \otimes l_2\} S\{l_2 \otimes x\} - S\{l_1 \otimes y\}}{(S\{l_1 \otimes l_2\})^2 - 1}$, $\beta = \frac{S\{l_1 \otimes l_2\} S\{l_1 \otimes y\} - S\{l_2 \otimes x\}}{(S\{l_1 \otimes l_2\})^2 - 1}$, and $\gamma = \frac{||x||^2 - \alpha^2 - \beta^2 - 2\alpha\beta S\{l_1 \otimes l_2\}}{||V\{l_1 \otimes l_2\}||^2}$. Then, we can calculate θ_1 and θ_2 using the sub-problem 1.

To solve the sub-problem 3, we have $||y - q \otimes x \otimes q^*|| = ||\gamma||$ as the general movement equation, where $\theta_0 = arctan2(S\{l \otimes x^{'} \otimes y^{'}\}, S\{x^{'} \otimes y^{'}\})$; then $\theta = \theta_0 \pm cos^{-1}\left(\frac{||x^{'}||^2 + ||y^{'}||^2 - \gamma^{'2}}{2||x^{'}||||y^{'}||}\right)$, where $x' = x + S\{l \otimes x\}l$, $y' = y + S\{l \otimes y\}l$, $\gamma^{'2} = \gamma^2 + |S\{l \otimes (a - b)\}|$.

For forward kinematics, given the six variable joints $(q_1, q_2, q_3, q_4, q_5, q_6)$, we obtain the coordinates of end effector (x, y, z) with respect to the base of the KC.

As Equation (1), the transformation operators DQ can be obtained as follows:

$$\hat{q}_i = (\hat{q}_{S_i}, \hat{q}_{V_i}) \qquad or \qquad \hat{q}_i = q + \varepsilon \hat{q}_i^0, \tag{7}$$

where for prismatic joints $q_i = [1, 0, 0, 0]$ and $q_i^0 = [0, q_1^0, q_2^0, q_3^0]$, for revolute joints $q_i = [\cos(\frac{\theta_i}{2}), \sin(\frac{\theta_i}{2})d_i]$ and $q_i^0 = \frac{1}{2}(p_i - q_i \otimes p_i \otimes q_i^*) \otimes q_i$ or $q_i^0 = [0, \sin(\frac{\theta_i}{2})m_i]$. In addition, where d_i is the rotation axis vector, m_i is the vector moment, and θ_i is the angle of rotation and $i = 1, 2, ...n$.

The general rigid body transformation operation is given by:

$$\hat{q}_{1n} = \hat{q}_1 \odot \hat{q}_2 \odot \hat{q}_3 \odot \cdots \odot \hat{q}_n, \tag{8}$$

where $\hat{q}_{1n} = q_{1n} + \varepsilon \hat{q}_s^0$. $\hat{q}_{1n}$ transform vectors and positions from 1 to n.

The orientation and position of the end effector can be found as follows: $\hat{l}_n = l_n + \varepsilon l_n^0$ and $\hat{l}_{n-1} = l_n + \varepsilon l_{n-1}^0$ are the representations of the Plücker coordinates n^{th} and $(n-1)^{th}$, respectively. We also have $\hat{l}_n' = l_n' + \varepsilon l_n'^0 = \hat{q}_{1n} \odot \hat{l}_n \odot \hat{q}_{1n}^*$, and $\hat{l}_{n-1}' = l_{n-1}' + \varepsilon l_{n-1}'^0 = \hat{q}_{1n-1} \odot \hat{l}_{n-1} \odot \hat{q}_{1n-1}^*$ are the representations of the Plücker after transformation. The orientation of the end effector is $\hat{l}_6'$. The end effector position can be found as follows:

$$\begin{aligned} p_n = (V\{R\{\hat{q}_{1n} \odot \hat{l}_n \odot \hat{q}_{1n}^*\}\}) \times (V\{D\{\hat{q}_{1n} \odot \hat{l}_n \odot \hat{q}_{1n}^*\}\}) \\ +(((V\{R\{\hat{q}_{1n-1} \odot \hat{l}_{n-1} \odot \hat{q}_{1n-1}^*\}\}) \times (V\{D\{\hat{q}_{1n-1} \odot \hat{l}_{n-1} \odot \hat{q}_{1n-1}^*\}\})) \\ \cdot(V\{R\{\hat{q}_{1n} \odot \hat{l}_n \odot \hat{q}_{1n}^*\}\})) * (V\{R\{\hat{q}_{1n} \odot \hat{l}_n \odot \hat{q}_{1n}^*\}\}). \end{aligned} \tag{9}$$

We can obtain Equation (9) using the intersection of two orthogonal unit line vectors given by:

$$\mathbf{r} = d_b \times m_b + (d_a \times m_a \cdot d_b)d_b, \tag{10}$$

$$\mathbf{r} = d_a \times m_a + (d_b \times m_b \cdot d_a)d_a. \tag{11}$$

For inverse kinematics, given the coordinates of end effector, p_6, and the orientation, $\hat{l}_6'$, in Euler parameters, $(x, y, z, \phi, \theta, \psi)$, we can obtain the six variable joints, $(q_1, q_2, q_3, q_4, q_5, q_6)$, as we show below.

We have as input parameters:

$$\hat{q}_{in} = \begin{bmatrix} q_{in} \\ q_{in}^0 \end{bmatrix} = \begin{bmatrix} \hat{l}_6' \\ p_6 \end{bmatrix}, \tag{12}$$

where $q_{in} = [q_0, q_1, q_2, q_3]$, end effector orientation, is a real part of dual quaternion $\hat{q}_{in}$. In addition, $q_{in}^0 = [q_0^0, q_1^0, q_2^0, q_3^0]$, end effector position, and dual part of dual quaternion $\hat{q}_{in}$.

We have then:

$$\hat{l}_6' = R\{\hat{q}_{16} \odot \hat{l}_6 \odot \hat{q}_{16}^*\} = q_{in}, \tag{13}$$

$$\begin{aligned} p_6 = (V\{R\{\hat{q}_{16} \odot \hat{l}_6 \odot \hat{q}_{16}^*\}\}) \times (V\{D\{\hat{q}_{16} \odot \hat{l}_6 \odot \hat{q}_{16}^*\}\}) \\ +(((V\{R\{\hat{q}_{15} \odot \hat{l}_5 \odot \hat{q}_{15}^*\}\}) \times (V\{D\{\hat{q}_{15} \odot \hat{l}_5 \odot \hat{q}_{15}^*\}\})) \\ \cdot(V\{R\{\hat{q}_{16} \odot \hat{l}_6 \odot \hat{q}_{16}^*\}\})) * (V\{R\{\hat{q}_{16} \odot \hat{l}_6 \odot \hat{q}_{16}^*\}\}) = q_{inV}^0. \end{aligned} \tag{14}$$

An inverse kinematics problem has been solved using the appropriate problems of Paden–Kahan.

Wrist position depends only for the first three joints and wrist orientation depends for the rest of the joints. For this reason, the first joint to calculate is θ_3. We define two points, the first point p_w allocated on the intersections of axis 5 and 6, and the second point p_b on the intersection of axes 1 and 2:

$$\begin{aligned} (V\{R\{\hat{q}_{13} \odot \hat{l}_6 \odot \hat{q}_{13}^*\}\}) \times (V\{D\{\hat{q}_{13} \odot \hat{l}_6 \odot \hat{q}_{13}^*\}\}) \\ +(((V\{R\{\hat{q}_{13} \odot \hat{l}_5 \odot \hat{q}_{13}^*\}\}) \times (V\{D\{\hat{q}_{13} \odot \hat{l}_5 \odot \hat{q}_{13}^*\}\})) \\ \cdot(V\{R\{\hat{q}_{13} \odot \hat{l}_6 \odot \hat{q}_{13}^*\}\})) * (V\{R\{\hat{q}_{13} \odot \hat{l}_6 \odot \hat{q}_{13}^*\}\}) = q_{inV}^0, \end{aligned} \tag{15}$$

$$\begin{aligned}
&(V\{R\{\hat{q}_{12}\odot\hat{l}_2\odot\hat{q}_{12}^*\}\})\times(V\{D\{\hat{q}_{12}\odot\hat{l}_2\odot\hat{q}_{12}^*\}\})\\
&+(((V\{R\{\hat{q}_{12}\odot\hat{l}_1\odot\hat{q}_{12}^*\}\})\times(V\{D\{\hat{q}_{12}\odot\hat{l}_1\odot\hat{q}_{12}^*\}\}))\\
&\cdot(V\{R\{\hat{q}_{12}\odot\hat{l}_2\odot\hat{q}_{12}^*\}\}))*(V\{R\{\hat{q}_{12}\odot\hat{l}_2\odot\hat{q}_{12}^*\}\})=q_b.
\end{aligned}\tag{16}$$

Doing the subtraction of both points and using the property of the distance between two preservation points by the rigid movements, we obtain sub-problem 3 of Paden–Kahan. The parameters of this sub-problem are:

$$\begin{aligned}
a&=(V\{R\{\hat{l}_6\}\})\times(V\{D\{\hat{l}_6\}\})+(((V\{R\{\hat{l}_5\}\})\times(V\{D\{\hat{l}_5\}\}))\cdot(V\{R\{\hat{l}_6\}\}))*(V\{R\{\hat{l}_6\}\})\\
b&=(V\{R\{\hat{l}_2\}\})\times(V\{D\{\hat{l}_2\}\})+(((V\{R\{\hat{l}_1\}\})\times(V\{D\{\hat{l}_1\}\}))\cdot(V\{R\{\hat{l}_2\}\}))*(V\{R\{\hat{l}_2\}\}),
\end{aligned}\tag{17}$$

and where l is the joint 3 and $\delta=q_{in}^0-p_b$. Using these parameters and using the sub-problem 3, we can find θ_3.

If we know θ_3 in Equation (15), we can obtain:

$$\begin{aligned}
&(V\{R\{\hat{q}_{12}\odot\hat{l}_6\odot\hat{q}_{12}^*\}\})\times(V\{D\{\hat{q}_{12}\odot\hat{l}_6\odot\hat{q}_{12}^*\}\})\\
&+(((V\{R\{\hat{q}_{12}\odot\hat{l}_5\odot\hat{q}_{12}^*\}\})\times(V\{D\{\hat{q}_{12}\odot\hat{l}_5\odot\hat{q}_{12}^*\}\}))\\
&\cdot(V\{R\{\hat{q}_{12}\odot\hat{l}_6\odot\hat{q}_{12}^*\}\}))*(V\{R\{\hat{q}_{12}\odot\hat{l}_6\odot\hat{q}_{12}^*\}\})=q_{inV}^0,
\end{aligned}\tag{18}$$

where $\hat{l}'_6=\hat{q}_3\odot\hat{l}_6\odot\hat{q}_3^*$ and $\hat{l}'_5=\hat{q}_3\odot\hat{l}_5\odot\hat{q}_3^*$.

With Equation (18), we obtain the sub-problem 2 of Paden–Kahan, where the parameters are:

$$a=(V\{R\{\hat{l}'_6\}\})\times(V\{D\{\hat{l}'_6\}\})+(((V\{R\{\hat{l}'_5\}\})\times(V\{D\{\hat{l}'_5\}\}))\cdot(V\{R\{\hat{l}'_6\}\}))*(V\{R\{\hat{l}'_6\}\}),\tag{19}$$

and where l_1 is the joint 1, d_1; parameter l_2 is the joint 2, d_2; value $b=q_{in}^0$. With these parameters and the sub-problem 2, we can found θ_1 and θ_2.

To find the angles of the wrist, we have to consider a new point $p_i=p_6+\lambda d_6$, initial point, allocated over joint axes d_6. To find the final point p_e, we need two imaginary axes so that this point is the position of point p_i after rotation of angles θ_4 and θ_5. Point p_i is the intersection of imaginary axes. This leads us to:

$$\begin{aligned}
&(V\{R\{\hat{q}_{45}\odot\hat{l}'_8\odot\hat{q}_{45}^*\}\})\times(V\{D\{\hat{q}_{45}\odot\hat{l}'_8\odot\hat{q}_{45}^*\}\})\\
&+(((V\{R\{\hat{q}_{45}\odot\hat{l}'_7\odot\hat{q}_{45}^*\}\})\times(V\{D\{\hat{q}_{45}\odot\hat{l}'_7\odot\hat{q}_{45}^*\}\}))\\
&\cdot(V\{R\{\hat{q}_{45}\odot\hat{l}'_8\odot\hat{q}_{45}^*\}\}))*(V\{R\{\hat{q}_{45}\odot\hat{l}'_8\odot\hat{q}_{45}^*\}\})=q_{in}^0+\lambda d_6,
\end{aligned}\tag{20}$$

where $\hat{l}'_8=\hat{q}_{13}\odot\hat{l}_8\odot\hat{q}_{13}^*$ and $\hat{l}'_7=\hat{q}_{13}\odot\hat{l}_7\odot\hat{q}_{13}^*$. Equation (20) provides the sub-problem 2 of Paden–Kahan. The parameters are:

$$a=(V\{R\{\hat{l}'_8\}\})\times(V\{D\{\hat{l}'_8\}\})+(((V\{R\{\hat{l}'_7\}\})\times(V\{D\{\hat{l}'_7\}\}))\cdot(V\{R\{\hat{l}'_8\}\}))*(V\{R\{\hat{l}'_8\}\}),\tag{21}$$

and where parameter l_1 is the imaginary axis 7, d_7; parameter l_2 is the imaginary axe 8, d_8; value $b=q_{in}^0+\lambda d_6$. With this parameters and the sub-problem 2, we can find θ_4 and θ_5.

To find the last parameter, θ_6, we need a point allowed over the last axis. We define $p_d=p_5+\lambda d_5$. We use two virtual axes to find the point p'_d that is the position of the point p_d after rotation of θ_6. Analogously to the above equations and the five angles known, we obtain:

$$\begin{aligned}
&(V\{R\{\hat{q}_6\odot\hat{l}'_{10}\odot\hat{q}_6^*\}\})\times(V\{D\{\hat{q}_6\odot\hat{l}'_{10}\odot\hat{q}_6^*\}\})\\
&+(((V\{R\{\hat{q}_6\odot\hat{l}'_9\odot\hat{q}_6^*\}\})\times(V\{D\{\hat{q}_6\odot\hat{l}'_9\odot\hat{q}_6^*\}\}))\\
&\cdot(V\{R\{\hat{q}_6\odot\hat{l}'_{10}\odot\hat{q}_6^*\}\}))*(V\{R\{\hat{q}_6\odot\hat{l}'_{10}\odot\hat{q}_6^*\}\})=q_{in}^0+\lambda d_6,
\end{aligned}\tag{22}$$

where $\hat{l}'_{10} = \hat{q}_{15} \odot \hat{l}_{10} \odot \hat{q}^*_{15}$ and $\hat{l}'_9 = \hat{q}_{15} \odot \hat{l}_9 \odot \hat{q}^*_{15}$. Equation (22) allows us to sub-problem 1. The parameters are:

$$a = (V\{R\{\hat{l}'_{10}\}\}) \times (V\{D\{\hat{l}'_{10}\}\}) + (((V\{R\{\hat{l}'_9\}\}) \times (V\{D\{\hat{l}'_9\}\})) \cdot (V\{R\{\hat{l}'_{10}\}\})) * (V\{R\{\hat{l}'_{10}\}\}), \quad (23)$$

and where parameter l is the imaginary axis 6, d_6; value $b = q^0_{in} + \lambda d_5$. With these parameters and the sub-problem 1, we can find θ_6.

We use inverse kinematics because we can obtain the base of our KC (shoulders or hips), and where the final effector and the orientation (hands and feet) are; thus, we have these parameters: $(x, y, z, \phi, \theta, \psi)$ and, using inverse kinematics, we obtain the six variable joints,$(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$, and use them to know where the elbow or knee are located.

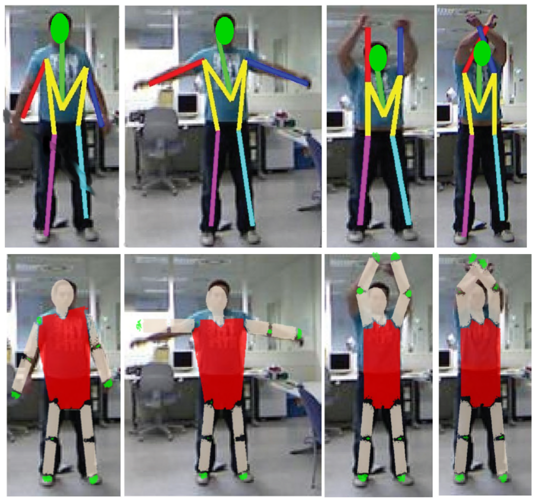


Figure 6. Results of our method after inverse kinematics (IK). The second row shows the model and joints being inferred (elbows and knees).

3. Results

- **3D Camera Calibration:** Our method works with any RGB-D sensor after correct calibration. In our experiments, we use a kinect device and calibrate the intrinsic and extrinsic parameters of the monocular and IR sensors. The calibration system is done similarly to [43] or [44,45].
- **Data sets:** To train and test our method, we use a combination of videos from our own data set and a subset of the publicly available CAD60 data set [46].
- **CAD60 data set:** The original CAD60 data set [46] contains 60 RGB-D videos, four subjects (two male, two female), four different environments (office, bedroom, bathroom and living room) and 12 different activities. This data set was originally created for the activity recognition task [47–49]. The size of images is 320 × 240 pixels.
- **Our data set:** It consists of seven videos with only one person on the scene moving his arms and legs. We had almost 1000 frames of people to obtain specific movements, e.g., crossing arms over one's body, to complement the CAD60 data set. Images were taken indoors in different scenarios. The subject inside the images is a male who wears different clothes. The size of the images is 320 × 240 pixels.

The ground truth of the joints in this data set was obtained by recording predictions from kinect. Thus, in order to make a fair comparison of the predictions from the methods being tested, we provide the videos to our human annotators to manually record the ground truth of the joint positions in the CAD60 data set. Thus, our annotators recorded over 15,000 frames of videos that correspond to 16 videos from the CAD60 data set with different activities and environments. For training and testing purposes, we use two different splits of such annotations. We chose to manually annotate the CAD60 data set because, to our knowledge, there is no RGBD data set with the ground truth of human pose joints. We will also publicly release our annotated videos for the benefit of the research community.

We can find some other data sets using RGB and depth images for pose estimation, but they can not be used in our proposed method due to annotation problems.

Metrics: The metrics we use in our different experiments are the probability of a correct kypoint (PCK), the average precision keypoint (APK) and error distance.

PCK: The probability of a correct keypoint (PCK) was introduced by Yang and Ramanan [1]. Given the bounding box, a pose estimation algorithm must report back the keypoint locations for body joints. The overlap between the keypoint bounding boxes was measured, which can suffer from quantization artifacts for small bounding boxes. A keypoint is considered correct if it lies within $\alpha \cdot max(h, w)$ of the ground truth bounding box, where h corresponds to the height and w to the width of the corresponding bounding box. α is a parameter that controls the relative threshold to consider the correctness of the keypoint.

APK: In a real system, however, one has no access to annotated bounding boxes at the test time, and one must also address the detection problem. One can cleanly combine the two problems by thinking of body parts (or rather joints) as objects to be detected, and evaluate object detection accuracy with a precision–recall curve. The average precision keypoint is another metric introduced by Yang and Ramanan [1], where, unlike PCK, it penalizes false-positives. Correct keypoints are also determined through the $\alpha \cdot max(h, w)$ relationship.

Error distance: This metric calculates the distance between the results and the correct labeled point. To do this, we calculate the distance error between the predicted result and the ground truth location. For each joint, we obtain an error score that is the mean value calculated from all of the frames.

3.1. Quantitative Results

Table 1 compares our results with Yang and Ramanan's [1] original method (Yang*) trained with the same images that we used to train our proposed method (P. Method*). Observing the results obtained in Table 1, and by comparing our proposed method with the original DPM, trained both with the same range of images and tested with the same range of images, but a different one of trained images, we have improved the results with the proposed method by adding depth information, a Kalman filter and using Denavit–Hartenberg (DH), in order to cut the number of points modeled in the DPM. Observing the results in Table 1, and independently of the data set used to test or train parts, our proposed method obtains better solutions. This means that the results can be repeatable with different data sets.

Table 1. Experimental comparisons with the state-of-the-art methods on our proposed data set. The probability of a correct kypoint (PCK) and the average precision keypoint (APK) metrics are expressed on %. Error is expressed in pixels.

Model	Metric	Head	Shoulders	Wrist	Hip	Ankle	Avg
Yang* [1]	APK	91.20	92.30	82.70	86.60	83.50	87.26
	PCK	91.50	89.00	85.80	89.90	83.80	88.00
	Error	8.17	8.81	10.87	9.37	11.59	9.76
P. Method* with KF with DH	APK	**97.50**	**98.30**	**92.20**	**94.70**	**94.00**	**95.34**
	PCK	**96.40**	**95.20**	**93.70**	**96.50**	**94.20**	**95.20**
	Error	**5.82**	**5.71**	**7.43**	**6.37**	**6.61**	**6.38**

Table 1 shows the results using KF and DH. , and using DQ will obtain the same results in accuracy as using DH. For this reason, a table comparing the original DPM model with our proposed method is not shown. We discuss in the next section the difference between DH and DQ.

3.2. Time Complexity Analysis

For our experiments, we use a system based on Windows 7 (Microsoft, Redmond, WA, USA) with 64 bits and 4 GB RAM. The processor used is Inter Core Quad 2.33 GHz (Intel Corporation, Santa Clara, CA, USA). We calculate for each frame the average time taken for the proposed algorithm to process the frame. The images used have 320 × 240 pixels.

In our previous work, we used (DH) kinematics instead of dual quaternions. Dual quaternions are faster than a transformation matrix used in DH and do not have singularities in their solutions.

Table 2. Number of operations between Denavit–Hartenberg and dual quaternions.

Method	Memory	Products	Sum/Subtract	Total
Homogeneous Matrix	16	64	48	112
Dual Quaternions	8	48	40	88

Table 2 shows the number of operations for one degree of freedom, for n degrees of freedom, we have on DH 64 $(n-1)$ products operations and 48 $(n-1)$ between sums and subtraction operations, while, for DQ. we have 48 $(n-1)$ products operations and 40 $(n-1)$ between sums and subtractions operations. Figure 7 shows how many operations we need for each degree of freedom added.

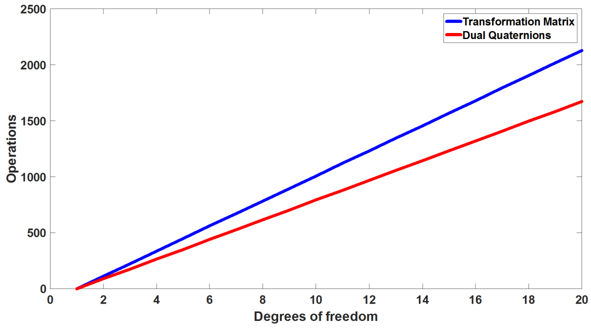


Figure 7. Comparing the number of operations between Denavit–Hartemberg and dual quaternions.

Figure 8 shows the time needed to make the operations. We can see that DQ is faster than DH; for this reason, we opted to use DQ.

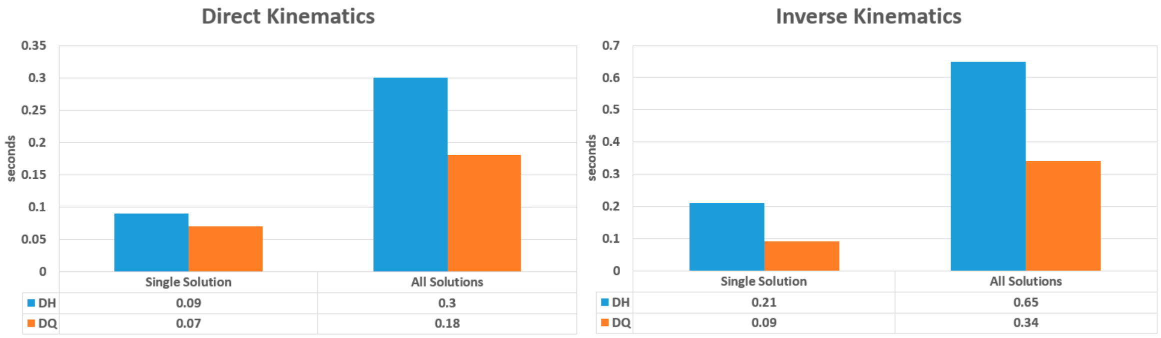


Figure 8. Computational time used.

All of these comparisons about computational cost using dual quaternions are for one kinematic chain. In our proposed method, we are using four kinematic chains for which we have to multiply these results by 4.

Finally, the computational cost of the proposed method is 6.85 s and the original DPM method takes 9.21 s.

4. Conclusions

In this paper, we have presented a 4D-DPM model using RGB-D information to improve accuracy and timing cost. We use MSER for foreground subtraction. We use dual quaternions to reduce the number of points of interest inside the imagery. We use a polisphere to draw the results and detect collisions between the different parts of the body. All of this allows us to reduce the time complexity during the training part using a smaller fraction of training samples.

Acknowledgments: This work was partially financed by Plan Nacional de Investigación y Desarrollo (I+D), Comision Interministerial de Ciencia y Tecnología (FEDER-CICYT) under the project DPI2013-44227-R.

Author Contributions: Enrique Martinez–Berti and Antonio–José Sánchez–Salmerón proposed the new method. Carlos Ricolfe–Viala contributed with calibration process. Enrique Martinez–Berti, Antonio–José Sánchez–Salmerón and Carlos Ricolfe–Viala conceived and designed the experiments; Enrique Martinez–Berti performed the experiments; Enrique Martinez–Berti and Antonio–José Sánchez–Salmerón analyzed the data; Enrique Martinez–Berti and Antonio–José Sánchez–Salmerón wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PS	Pictorial Structures
MRFs	Markov Random Fields
DPM	Deformable Parts Model
MSER	Maximally Stable Extremal Regions
PCK	Probability of a Correct Kypoint
APK	Average Precision Keypoint
KF	Kalman Filter
DQ	Dual Quaternions
KC	Kinematic Chains

References

1. Yang, Y.; Ramanan, D. Articulated human detection with flexible mixtures of parts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2878–2890.
2. Wang, F.; Li, Y. Beyond physical connections: Tree models in human pose estimation. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 596–603.
3. Pishchulin, L.; Andriluka, M.; Gehler, P.; Schiele, B. Poselet conditioned pictorial structures. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 588–595.
4. Toshev, A.; Szegedy, C. Deeppose: Human pose estimation via deep neural networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1653–1660.
5. Ramakrishna, V.; Munoz, D.; Hebert, M.; Bagnell, J.A.; Sheikh, Y. Pose Machines: Articulated Pose Estimation via Inference Machines. In *Computer Vision–ECCV 2014*; Springer: Berlin, Germany, 2014; pp. 33–47.

6. Shotton, J.; Girshick, R.; Fitzgibbon, A.; Sharp, T.; Cook, M.; Finocchio, M.; Moore, R.; Kohli, P.; Criminisi, A.; Kipman, A.; et al. Efficient human pose estimation from single depth images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2821–2840.
7. Martinez, E.; Nina, O.; Sanchez, A.; Ricolfe, C. Optimized 4D-DPM for Pose Estimation on RGBD Channels using polisphere models. In Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Porto, Portugal, 27 February–1 March 2017; Volume 5, pp. 281–288.
8. Martinez, E.; Sanchez-Salmeron, A.J.; Ricolfe-Viala, C. 4D-DPM model for pose estimation using Kalman filter constraints. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1–13.
9. Fischler, M.A.; Elschlager, R.A. The representation and matching of pictorial structures. *IEEE Trans. Comput.* **1973**, *22*, 67–92.
10. Eichner, M.; Ferrari, V. Better appearance models for pictorial structures. In Proceedings of the British Machine Vision Conference (BMVC), London, UK, 8–10 September 2009; Volume 2, p. 5.
11. Andriluka, M.; Roth, S.; Schiele, B. Pictorial structures revisited: People detection and articulated pose estimation. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 1014–1021.
12. Huang, C.M.; Chen, Y.R.; Fu, L.C. Visual tracking of human head and arms using adaptive multiple importance sampling on a single camera in cluttered environments. *IEEE Sens. J.* **2014**, *14*, 2267–2275, doi:10.1109/JSEN.2014.2309256.
13. Ning, X.; Guo, G. Assessing spinal loading using the kinect depth. *IEEE Sens. J.* **2013**, 13, 1139–1140, doi:10.1109/JSEN.2012.2230252.
14. Sapp, B.; Taskar, B. Modec: Multimodal decomposable models for human pose estimation. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 3674–3681.
15. Wang, Y.; Tran, D.; Liao, Z.; Forsyth, D. Discriminative hierarchical part-based models for human parsing and action recognition. *J. Mach. Learn. Res.* **2012**, *13*, 3075–3102.
16. Bourdev, L.; Malik, J. Poselets: Body part detectors trained using 3d human pose annotations. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009; pp. 1365–1372.
17. Ionescu, C.; Li, F.; Sminchisescu, C. Latent structured models for human pose estimation. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2220–2227.
18. Gkioxari, G.; Arbeláez, P.; Bourdev, L.; Malik, J. Articulated pose estimation using discriminative armlet classifiers. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 3342–3349.
19. Grest, D.; Woetzel, J.; Koch, R. Nonlinear body pose estimation from depth images. In *Pattern Recognition*; Springer: Berlin, Germany, 2005; pp. 285–292.
20. Plagemann, C.; Ganapathi, V.; Koller, D.; Thrun, S. Real-time identification and localization of body parts from depth images. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–8 May 2010; pp. 3108–3113.
21. Helten, T.; Baak, A.; Bharaj, G.; Muller, M.; Seidel, H.P.; Theobalt, C. Personalization and Evaluation of a Real-Time Depth-Based Full Body Tracker. In Proceedings of the 2013 International Conference on 3D Vision, Seattle, WA, USA, 23 June–1 July 2013, pp. 279–286.
22. Baak, A.; Müller, M.; Bharaj, G.; Seidel, H.P.; Theobalt, C. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Consumer Depth Cameras for Computer Vision*; Springer: Berlin, Germany, 2013; pp. 71–98.
23. Spinello, L.; Arras, K.O. People detection in RGB-D data. In Proceedings of the 2011 IEEE Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011.
24. Ganapathi, V.; Plagemann, C.; Koller, D.; Thrun, S. Real time motion capture using a single time-of-flight camera. In Proceedings of the 2010 IEEE Conference Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 755–762.

25. Ye, M.; Yang, R. 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 (CVPR), Columbus, OH, USA, 23–28 June 2014.
26. Ding, M.; Fan, G. Articulated and Generalized Gaussian Kernel Correlation for Human Pose Estimation. *IEEE Trans. Image Process.* **2016**, *25*, doi:10.1109/TIP.2015.2507445 .
27. Ganapathi, V.; Plagemann, C.; Koller, D.; Thrun, S. Real-time human pose tracking from range data. In Proceedings of the 12th European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012.
28. Ganapathi, V.; Plagemann, C.; Koller, D.; Thrun, S. Real time motion capture using a single time-of-flight camera. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010.
29. Baak, A.; Muller, M.; Bharaj, G.; Seidel, H.; Theobalt, C. A datadriven approach for real-time full body pose reconstruction from a depth camera. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011.
30. Ye, M.; Wang, X.; Yang, R.; Ren, L.; Pollefeys, M. Accurate 3D pose estimation from a single depth image. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011.
31. Wei, X.; Zhang, P.; Chai, J. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph.* **2012**, *31*, 188.
32. Stoll, C.; Hasler, N.; Gall, J.; Seidel, H.; Theobalt, C. Fast articulated motion tracking using a sums of Gaussians body model. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011.
33. Ding, M. Fast human pose tracking with a single depth sensor using sum of Gaussians models. *Adv. Visual Comput.* **2014**, 8887, 599–608.
34. Ding, M.; Fan, G. Generalized sum of Gaussians for real-time human pose tracking from a single depth sensor. In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 5–9 January 2015.
35. Taylor, J.; Shotton, J.; Sharp, T.; Fitzgibbon, A. The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
36. Kurmankhojayev, D.; Hasler, N.; Theobalt, C. Monocular pose capture with a depth camera using a Sums-of-Gaussians body model. *Pattern Recognit.* **2013**, *8142*, 415–424.
37. Sridhar, S.; Rhodin, H.; Seidel, H.; Oulasvirta, A.; Theobalt, C. Real-time hand tracking using a sum of anisotropic Gaussians model. In Proceedings of the International Conference on 3D Vision (3DV), Tokyo, Japan, 8–11 December 2014.
38. Tsin, Y.; Kanade, T. A correlation-based approach to robust point set registration. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2004.
39. Lai, K.; Bo, L.; Ren, X.; Fox, D. Detection-based object labeling in 3D scenes. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), St Paul, MN, USA, 14–18 May 2012; pp. 1330–1337.
40. Sridhar, S.; Oulasvirta, A.; Theobalt, C. Interactive markerless articulated hand motion tracking using RGB and depth data. In Proceedings of the International Conference on Computer Vision (ICCV) 2013, Sydney, Australia, 1–8 December 2013.
41. Matas, J.; Chum, O.; Urban, M.; Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **2004**, *22*, 761–767.
42. Martinez, E.; Sanchez, A.; Ricolfe, C.; Nina, O. Human Pose Estimation for RGBD Imagery with Multi-Channel Mixture of Parts and Kinematic Constraints. *WSEAS Trans. Comput.* **2016**, *15*, 279–286.
43. Berti, E.M.; Salmerón, A.J.S.; Benimeli, F. Human-Robot Interaction and Tracking Using low cost 3D Vision Systems. *Romanian J. Tech. Sci. Appl. Mech.* **2012**, *7*, 1–15.
44. Ricolfe, C.; Sanchez, A.; Martinez, E. Calibration of a wide angle stereoscopic system. *Opt. Lett.* **2011**, *36*, 3064–3066.
45. Ricolfe, C.; Sanchez, A.; Martinez, E. Accurate calibration with highly distorted images. *Appl. Opt.* **2012**, *51*, 89–101.

46. Sung, J.; Ponce, C.; Selman, B.; Saxena, A. Human activity detection from RGBD images. *Plan Act. Intent Recognit.* **2011**, *64*, 47–55.
47. Wang, J.; Liu, Z.; Wu, Y. Learning actionlet ensemble for 3D human action recognition. In *Human Action Recognition with Depth Cameras*; Springer: Berlin, Germany, 2014; pp. 11–40.
48. Shan, J.; Akella, S. 3D Human Action Segmentation and Recognition using Pose Kinetic Energy. In Proceedings of the 2014 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), Evanston, IL, USA, 11–13 September 2014.
49. Faria, R.D.; Premebida, C.; Nunes, U. A Probalistic Approach for Human Everyday Activities Recognition using Body Motion from RGB-D Images. In Proceedings of the 2014 RO-MAN: 23rd IEEE International Symposium on Robot and Human Interactive Communication, Edinburgh, UK, 25–29 August 2014.

© 2017 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).