Human Pose Estimation for RGBD Imagery with Multi-Channel Mixture of Parts and Kinematic Constraints

ENRIQUE MARTINEZ-BERTI
Universitat Politecnica de Valencia
Instituto AI2
Camino de Vera s/n, Valencia
SPAIN
enmarbe1@etsii.upv.es

ANTONIO J. SNCHEZ-SALMERN
Universitat Politecnica de Valencia
Instituto AI2
Camino de Vera s/n, Valencia
SPAIN
asanchez@isa.upv.es

CARLOS RICOLFE-VIALA
Universitat Politecnica de Valencia
Instituto AI2
Camino de Vera s/n, Valencia
SPAIN
cricolfe@isa.upv.es

OLIVER NINA
Univeristy of Center Florida
Center for Research in Computer Vision
Scorpius St, 4328 Orlando
UNITED STATES
olivernina@gmail.com

MUBARAK SHAH
Univeristy of Center Florida
Center for Research in Computer Vision
Scorpius St, 4328 Orlando
UNITED STATES
shah@crcv.ucf.edu

Abstract: In this paper, we present a approach that combines monocular and depth information with a multi-channel mixture of parts model that is constrained by a structured linear quadratic estimator for more accurate estimation of joints in human pose estimation. Furthermore, in order to speed up our algorithm, we introduce an inverse kinematics optimization that allows us to infer additional joints that were not included in the original solution. This allows us to train in less time and with only a subset of the total number of joints in the final solution. Our results show a significant improvement over state of the art methods on the CAD60 and our own dataset. Also, our method can be trained in less time and with smaller fraction of training samples when compared to state of the art methods.

Key–Words: DPM, Kalman Filter, Pose Estimation, Kinematic Constraints

1 Introduction

Human pose estimation has been extensively studied for many years in computer vision. There have been many attempts to improve human pose estimation with methods that work mainly with monocular RGB images such as [33, 30, 15, 26, 18].

With the ubiquity and increased use of depth sensors, methods that use RGBD imagery are fundamental. One of the methods that use such imagery and that is currently considered the state of the art for human pose estimation is Shotton et al. [22], which was commercially developed for the Kinect device. Shotton's method allows real-time joint detection for human pose estimation based solely on depth channels.

Despite the state of the art performance of [22] and the comercial success of Kinect, many of the drawbacks of [22] make it difficult to adopt in any other type of 3D computer vision system.

Some of the drawbacks of [22] include copyright and licensing issues that restricts 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 thusands) required to train its deep random forest algorithm which could make training cumbersome.

Another drawback of [22], is that its model is trained only on depth information, thus discarding potentially important information that could be found in the RGB channels that could help approximate human pose more accurately.

To alleviate these and other drawbacks in [22], we propose a novel approach that takes advantage of both RGB and depth information combined in a multichannel mixture of parts for pose estimation in single frame images coupled with a skeleton constrained linear quadratic estimator that makes use of rigid information of a human skeleton to improve joint tracking in consecutive frames. In contrast to Kinect, our approach makes our model easily trainable even for non-human poses. Finally, in order to speed up the training process of proposed method, we propose an inverse kinematics optimization for inference of other joints not considered initially which reduces training time significantly.

The main contribution of our method extends to: (i) and optimized multi-channel mixture of parts model that allows detection of parts in RGBD images; (ii) a linear quadratic estimator that makes use of rigid information and connected joints of human pose; (iii) an optimization for unsolved joints through inverse kinematics that allows the model to be trained with fewer joints and in less time.

Our results show significant improvements over the state of the art in both the publicly available CAD60 dataset.

Related Work. Human pose estimation has been studied for many years and some of the methods in the literature that attempt to solve this problem date back to the use of Pictorial Structures (PS) introduced by [7]. More recent methods improve upon the concept of PS with improved features or inference models such as in [6, 1, 15].

Other methods that use more robust joint relationship include: [33] which uses a mixtures of parts model, [20] which uses a multimodel decomposable model, and [32] which considers a part-based models introducing hierarchical poselets. Other methods that have attempted to reconstruct 3D pose estimation from RGB monocular images include [4, 11, 8].

Object detection has been done using RGBD using MRFs and features from both RGB and depth [14].

Recently, 3D cameras such as Kinect have added a new dimension to computer vision problems. Such cameras allow us to capture not only RGB information as done with monocular cameras but also depth (D) information whose intensities depict an inversely proportional relationship of the distance of the objects to the camera.

Some methods that use depth images to reconstruct pose estimations include [9, 16, 22, 10, 2, 24]. Among such methods, Shotton et al. [22], which was developed for the Kinect algorithm, has become the state of the art for performing human pose estimation which predicts 3D positions of body joints from a single depth image.

2 Proposed Method

Section 2.1 explains the formulation of our four dimensional mixture of parts model. Section 2.2 explains our structured quadratic linear estimator for correcting joints in consecutive frames. Finally, section 2.3 describes the optimization of the computation complexity of our model.

2.1 Multi-channel Mixture of Parts

Until recently, Yang and Ramanan's method [33] had been a state of the art method for pose estimation in monocular images. However, Yang and Ramanan's method performs poorly on images that vary from those in its training set and even after retraining, the method only improves by a small margin.

Although there have been other algorithms that have improved upon Yang and Ramanan such as [30, 15, 18], all these methods, including Yang and Ramanan, use a mixture of parts for only the RGB dimension of channels. In contrast, in our method, we use 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**.

Hence, our method differs significantly from other previous methods in many important ways that we explain in this section. Furthermore, our implementation allows us to speed up training time by several factors, which will be described subsequently.

In our method, we formulate a score function (S) for the parts or joints that belong to pose through an appearance and deformation functions as follows:

$$S\left(I,x,t)\right) = \sum_{i\in V} \phi_i\left(I,x_i,t_i\right) + \sum_{ij\in E} \psi_{i,j}\left(I,x_i,t_i,x_i'\right), \quad (1)$$

where $x_i' = (x_j, t_j)$, I corresponds to the RGBD image, x is the location of the joint i which corresponds to the type of joint being detected, j is the potential joint being connected to i and $t = 1, \cdots, T$ is the the mixture component of joint i that expands to parts that have experienced different transformations such as rotation, translation, orientation and others. In contrast to [33], to improve training time in our method, our transformation function was implemented independently from the rest of the algorithm which allows us to speed training time in this step more than ten fold. The terms ϕ and ψ in equation 1 correspond to appearance and deformation models respectively. The appearance model calculates a score for the features of type assignment t_i whereas the deformation model provides a score for the deformation distance of type assignments t_i and t_j. These models are constrained with the tree structure represented by $G(V, E)$, where a vertex $i \in V$ represents a part and the edge $(i, j) \in E$ represents the co-occurrence of part i and j for optimization purposes since the computation of all the possible assignments is exponential.

In order to obtain features and deformations in all RGBD channels, we formulate ϕ and ψ as a multi-channel mixture of parts in the following way:

$$\phi_i\left(I,x_i,t_i\right) = \begin{bmatrix} \omega_{i\ m}^{t_i} \cdot \phi\left(I_m,x_i\right) + b_{i\ m}^{t_i} \\ \omega_{i\ d}^{t_i} \cdot \phi\left(I_d,x_i\right) + b_{i\ d}^{t_i} \end{bmatrix} \quad (2)$$

$$\psi_{ij}\left(I,x_i,t_i,x_j,t_j\right) = \begin{bmatrix} \omega_{ij\ \ m}^{t_i,t_j} \cdot \psi(x_i - x_j)_m + b_{ij\ \ m}^{t_it_j} \\ \omega_{ij\ \ d}^{t_i,t_j} \cdot \psi(x_i - x_j)_d + b_{ij\ \ d}^{t_it_j} \end{bmatrix}$$

where $\phi\left(I,x_i\right)$ is the appearance function represented by HOG [5] that extracts features from a

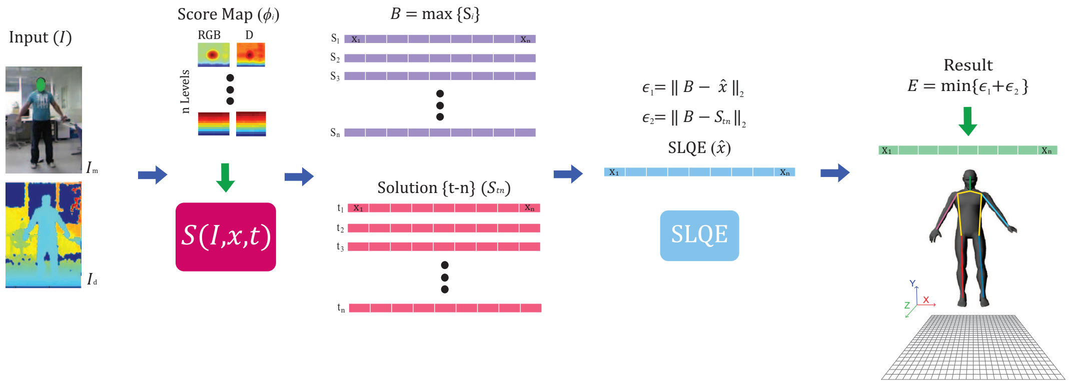

Figure 1: Outline of our method

monocular (I_m) or detph (I_d) images at pixel location x_i. $b_i^{t_i}$ is a parameter that corresponds to the assignment of part i in either channel, $b_{ij}^{t_i t_j}$ is another parameter that describes co-occurrence assignments of part i and j. Notice that in contrast to [33] the number of mixture parts in our equation 2 is twice as many for adding a depth channel. This extra number of mixture components is a complement to mixtures from RGB dimensions and allows to improve detection scores for all RGBD channels.

The deformation function is given by $\psi(x_i - x_j)_c = \begin{bmatrix} dx & dx^2 & dy & dy^2 \end{bmatrix}$, where $dx = x_i - x_j$ and $dy = y_i - y_j$, correspond to the location of part i with respect to j on image I_c for the respective type of image c.

Because the structure of $G(V, E)$ is a tree, we use dynamic programming to calculate S for each node in the tree with an extra second term as compared to [33] for calculating the scores and message passing in a way to accommodate for depth channels. Let $kids\,(i)$ be the set of children of part i in G. We compute the message part i that passes to its parent j in this way:

$$score_i\,(t_i, x_i) = b_i^{t_i} + \begin{bmatrix} \omega_{t_i m}^{i} \cdot \phi\,(I_m, p_i) \\ \omega_{t_i d}^{i} \cdot \phi\,(I_d, p_i) \end{bmatrix} + \sum_{k \in kids(i)} m_k\,(t_i, x_i) \tag{3}$$

$$m_i\,(t_j, x_j) = \max_{t_i} b_{ij}^{t_i, t_j} \max_{x_i} score\,(t_i, x_i) + \begin{bmatrix} w_{ij}^{t_i, t_j}{}_m \cdot \psi(x_i - x_j)_m \\ w_{ij}^{t_i, t_j}{}_d \cdot \psi(x_i - x_j)_d \end{bmatrix} \tag{4}$$

Equation 3 computes the local score of part i, at all pixel locations p_i and for all possible types t_i, by collecting messages from the children of i. Equation 4 computes every location and type of its child part i. Once messages are passed to the root $(i = 1)$, $score_1\,(c_1, x_1)$ represents the best scoring configuration for each root type and position.

In contrast to [33], we parameterize equation 1 as $S(I, x, t) = \alpha \cdot \Phi(I, x, t)$ and $\alpha = (w, b)$ to solve the following structural support vector machine (SVM) primal with the following conditions for processing positive and negative samples that allows us to solve the most violated constraint as independent steps i, thus improving training time when compared to [33].

$$\arg \min_{w, \xi \geq 0} \frac{1}{2} \alpha \cdot \alpha + C \sum_n \xi_n \tag{5}$$

$$\text{s.t. } \forall n \; \epsilon \text{ pos } \beta \cdot \Phi(I_{ni}, x_{ni}, t_{ni}) \geq 1 - \xi_{ni}$$

$$\forall n \; \epsilon \text{ neg}, \forall x_n, t_n \;\; \beta \cdot \Phi(I_n, x_n, t_n) \leq 1 - \xi_n$$

2.2 Joint Detection in Consecutive Frames

So far we have dealt only with pose estimation for every single frame independently, however, most joint movement performed in normal circumstances display uniform and constant change of displacement and velocity. Hence we can use the properties of velocity and acceleration of the joints in order to predict based on the past where the joints would most likely be. This motion-based prediction could help us validate our frame-based prediction.

One way to predict joint location based on previous detections is by using a linear quadratic estimator (LQE) [12]. Using a simple LQE works well when the points being tracked are independent from 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 which make the movement of one joint related to the one being connected. For instance, the movement of a foot joint would be relative to a parent joint such as a knee or a hip.

In order to utilize this joint relationship, we introduce a novel skeleton constrained linear quadratic estimator (SLQE) which uses joint relationships constraints from a human skeleton model to predict the location of all joints at the same time. In this section we explain this step of our approach.

We first define a state joint obtained by equation 1 with its respective vector components for position, velocity and acceleration as follows:

$$x'_i = \begin{bmatrix} x_i & y_i & vx_i & vy_i & ax_i & ay_i \end{bmatrix}^T$$

We also define the measurement matrix for a joint as H_1 that considers only the location component x_i and y_i of the joint.

$$H_1 = \begin{bmatrix} 1 & 0 & 0_{1\times4} \\ 0 & 1 & 0_{1\times4} \\ 0_{4\times1} & 0_{4\times1} & 0_{4\times4} \end{bmatrix} \tag{6}$$

Thus the measurement matrix for all joints is represented as :

$$H = \begin{bmatrix} H_1 & 0_{6\times6} & 0_{6\times6} & 0_{6\times6} \\ 0_{6\times6} & H_1 & 0_{6\times6} & 0_{6\times6} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{6\times6} & 0_{6\times6} & 0_{6\times6} & H_1 \end{bmatrix} \tag{7}$$

Given a state model A that models the relationship of each joint with respect to all other joints being consider, we define a pair of joints being connected to each other as A_1 and A_2 as:

$$A_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$A_2 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

Thus the final transition state matrix A for all the joints is defined as:

$$A = \begin{bmatrix} A_1 & A_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_1 & 0 & 0 & 0 & A_2 & 0 & 0 \\ 0 & 0 & A_1 & 0 & 0 & 0 & A_2 & 0 \\ 0 & 0 & A_2 & A_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_1 & A_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_2 & A_1 \end{bmatrix} \tag{10}$$

Notice that joints whose movement depends on another are paired up through the relationship $A_1 A_2$. Joints that are connected to each other have their movement dependent on each other thus their velocity and acceleration components are subtracted from each other.

The prediction of a posteriori joint $\mathbf{x} = [x'_1, \cdots, x'_n]$ at time t now depends on the structure embedded in A and can be calculated with:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} \tag{11}$$

We also calculate a posteriori error covariance P_t, such that:

$$P_t = AP_{t-1}A^T + Q \tag{12}$$

were Q is the measurement noise which in our case is an identity matrix.

We also compute the residual covariance S based on the noise covariance prediction R to calculate the gain K in this way:

$$S = HP_tH^T + R \tag{13}$$

$$K = P_tH^TS^{-1}$$

Once the outcome of the measurement $\mathbf{z}$ is observed, these estimates are updated using gain K, with more weight being given to estimates with higher certainty.

The final estimation of the coordinate joints by our SQLE is given by:

$$\hat{x} \; = \; H \cdot \mathbf{x}_{t-1} \tag{14}$$

Although for continuous movements SLQE can predict accurately the direction and speed of the movement, in cases were joint movement changes direction suddenly or there is increased noise such prediction could fail. To avoid this issue, we compare our prediction from SLQE and the last successful prediction from the last frame $B = \max_i S_{it}$, were S_i is the score function from 1 at frame t.

Thus we can avoid mistakes by SQLE or the score function by choosing the solution $\hat{x}$ or S_{t-1} with the least error $\min(\epsilon_1, \epsilon_2)$,

$$\epsilon_1 = \parallel B - \hat{x} \parallel_2 \tag{15}$$

$$\epsilon_2 = \parallel B - S_{t-1} \parallel_2$$

Because of the algorithm's recursive nature, this process can run in real time using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required.

3D Pose Estimation. Once the coordinates of the joints have been calculated in X and Y planes, finding the coordinates of such in the Z plane is as simple as converting the pixel values in the depth images back into Z coordinates.

2.3 Model Optimization.

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

In this section we explain an optimization technique that makes use of inverse kinematic equations in order to infer shoulder and knee joints by training our model with fewer parts.

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 detected in our previous step, inverse kinematics to obtain missing joints is calculated using Denavit-Hartemberg (D-H) notation [29, 13].

State Variables: The human body model is divided into 4 main kinematic chains (KC) that perform collision detection with their correspondent state variables, in essence: 1 KC for each arm and 1 for each leg.

D-H Optimization: To control each of the actuators in these model KC, we use D-H. In this sense, we use 6 joints for each KC for shoulders, hips, hands and feet.

First, we establish the base coordinate system (X_0, Y_0, Z_0) at the supporting base with Z_0 axis lying along the axis of motion of joint 1. 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 $X_i = \pm\left(Z_{i-1} \times Z_i\right) / \left\|Z_{i-1} \times Z_i\right\|$ 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 previous axis to the common normal), a_i (length of the common normal), α_i (angle of the common normal with respect to the new axis)

Given the 6 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. For inverse kinematics, given the coordinates of end effector and the orientation in euler parameters, $(x, y, z, \phi, \theta, \psi)$, we obtain the 6 variable joints, $(q_1, q_2, q_3, q_4, q_5, q_6)$.

Given the homogeneous transformation matrix that establishes the relationship of a joint with an adjacent one:

$$^{i-1}A_i(q_i) = \begin{bmatrix} c_\theta & -c_\alpha \cdot s_\theta & s_\alpha \cdot s_\theta & a_i \cdot c_\theta \\ s_\theta & c_\alpha \cdot c_\theta & -s_\alpha \cdot c_\theta & a_i \cdot s_\theta \\ 0 & s_\alpha & c_\alpha & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

where $s_\alpha = \sin(\alpha_i)$, $c_\alpha = \cos(\alpha_i)$, $s_\theta = \sin(\theta_i)$, $c_\theta = \cos(\theta_i)$. The location of the end effector relative to the reference can be obtained by the following relationship:

$$^0T_6\left(q_1, q_2, q_3, q_4, q_5, q_6\right) = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \cdot {}^4A_5 \cdot {}^5A_6$$

where $A_i = {}^{i-1}A_i(q_i)$. It is paramount to use geometric models for the first three joints, thus, we have the coordinates for the final effector (x, y, x) and after applying geometric models we can obtain the first three joints:

$$q_1 = \arctan\left(\frac{y}{x}\right) \quad (17)$$

$$q_3 = \arctan\left(\frac{\pm\sqrt{1 - \cos^2\left(\frac{x^2+y^2+z^2-a_2-a_3}{2\cdot a_2 \cdot a_3}\right)}}{\cos\left(\frac{x^2+y^2+z^2-a_2-a_3}{2\cdot a_2 \cdot a_3}\right)}\right) \quad (18)$$

$$q_2 = \arctan\left(\frac{z}{\pm\sqrt{x^2+y^2}}\right) - \varphi \quad (19)$$

where,

$$\varphi = -\arctan\left(\frac{a_3 \cdot \sin\left(\frac{x^2+y^2+z^2-a_2-a_3}{2\cdot a_2 \cdot a_3}\right)}{a_2 + a_3 \cdot \cos\left(\frac{x^2+y^2+z^2-a_2-a_3}{2\cdot a_2 \cdot a_3}\right)}\right)$$

Now we can use inverse kinematics to calculate the last three joints. We define $^0R_6 = {}^0R_3 \cdot {}^3R_6$ for the sub matrix rotation of $^0T_6$. We know the value of $^0R_6$ because is the orientation of the final effector and $^0R_3$ because is defined by $^0R_3 = {}^0R_1 \cdot {}^1R_2 \cdot {}^2R_3$ using (q_1, q_2, q_3). Then we calculate:

$$^3R_6 = [r_{ij}] = \left({}^0R_3\right)^{-1} {}^0R_6 \quad (20)$$

Applying $^3R_6 = {}^3R_4 \cdot {}^4R_5 \cdot {}^5R_6$ and using (q_4, q_5, q_6), we obtain the last three joints using equation 20.

$$q_4 = \arctan\left(\frac{r_{23}}{r_{13}}\right) \quad (21)$$

$$q_5 = \arccos\left(-r_{33}\right) \quad (22)$$

$$q_6 = \frac{\pi}{2} - \arctan\left(\frac{r_{32}}{r_{31}}\right) \quad (23)$$

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 an feet) are, thus we have these parameters: $(x, y, z, \phi, \theta, \psi)$ and using inverse kinematics, we obtain the 6 variable joints,$(q_1, q_2, q_3, q_4, q_5, q_6)$, and use them to know where the elbow or knee are located.

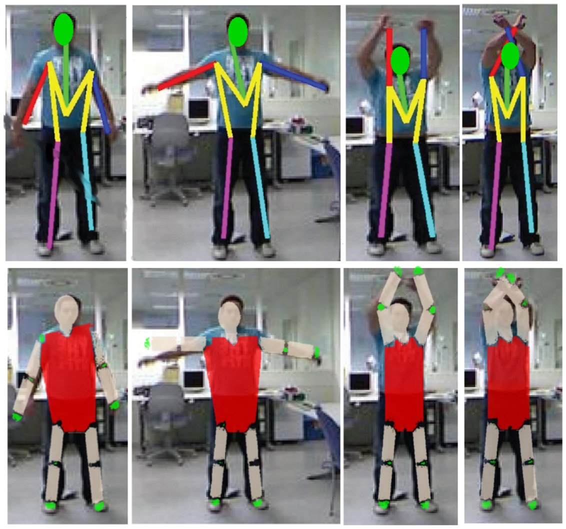

Figure 2: Results of our method after inverse kinematics (IK) optimization. Second row shows model and joints being inferred (elbows and knees)

3 Results.

3D Camera Calibration. Our method works with any RGBD sensor after the correct calibration, in our experiments we use a Kinect device and calibrate its intrinsic and extrinsic parameters of the monocular and IR sensors. The calibration system is done in a similar way to [3] or [27] and [28].

Datasets For training and testing of proposed method we use a subset of the publicly available CAD60 dataset [25].

CAD60 Dataset. The original CAD60 dataset [25] contains 60 RGB-D videos, 4 subjects (two male, two female), 5 different environments (office, bedroom, bathroom and living room) and 12 different activities. This dataset was originally created for the task of activity recognition [31, 21, 17].

Metrics. The metrics we use in our different experiments are PCK, APK and error distance.

PCK. The probability of correct keypoint (PCK) was introduced by Yang and Ramanan [33] where a keypoint is consider 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 corresponding bounding box. α is a paramter that controls the relative threshold for considering correctness of the keypoint.

APK. The average precision keypoint is another metric introduced by Yang and Ramanan [33] where in contrast to 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 the predicted result and the ground truth location. For each joint we obtain an error score which is the mean value calculated from all frames.

3.1 Quantitative Results.

Table 1 shows the results of comparing the proposed method (P. Method) with Yang and Ramanan [33] original method trained on the Image parse dataset [19] and also retrain it (Yang*) with the sames images that we trained our own model (P. Method*). Notice that although we retrain Yang and Ramanan's model our model is still significantly better than their method.

Model	Metric	Head	Shoul.	Wrist	Hip	Ank.	Avg
Yang [33]	APK	47.30	66.70	22.40	45.50	47.10	45.80
	PCK	62.50	70.40	39.00	60.50	57.9	58.06
	Error	15.53	12.23	22.34	16.29	18.50	16.97
Yang*[33]	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
Kinect [23]	APK	68.30	90.70	76.40	9.50	77.10	64.40
	PCK	79.50	94.40	85.00	23.50	85.9	73.66
	Error	13.17	6.85	9.64	18.42	11.28	15.87
P. Method	APK	**72.30**	**91.10**	**81.20**	**83.70**	**82.00**	**82.06**
	PCK	**83.60**	**95.00**	**88.70**	**87.30**	**89.20**	**88.76**
	Error	**9.95**	**6.81**	**8.73**	**8.58**	**8.40**	**8.49**
P. Method*	APK	**97.40**	**98.40**	**91.80**	**94.80**	**93.60**	**95.20**
	PCK	**96.20**	**94.90**	**94.0**	**97.40**	**93.60**	**95.22**
	Error	**5.95**	**5.81**	**7.25**	**5.02**	**5.40**	**5.89**

Table 1: Experimental comparisons with the state-of-the-art methods, and different components of our methods on CAD60 Dataset.

3.2 Time Complexity Analysis

For our experiments, we use a system with 4 GB RAM. We calculate for each frame the average time taken for our algorithm to process the frame. Our method takes about 7.26 seconds per frame whereas [33] takes about 9.21 seconds per frame which is approximately a 20% gain in performance from [33].

Although our time performance of our method is much slower than Kinect which is a real-time method, we have shown in our paper that our method can be trained with smaller number of frames as compared to Kinect which requires hundreds of thousands of frames.

4 Conclusions.

In this paper, we have presented a novel approach that combines monocular and depth information with a multi-channel mixture of parts model, a novel structured linear quadratic estimator and an inverse kinematics optimization for estimation of joints for human pose estimation in RGBD data.

Our results show a significant improvement over state of the art methods on the CAD60 and our own dataset. Also, our method can be trained in less time and with smaller fraction of training samples when compared to state of the art.

References:

[1] 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.

[2] 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.

[3] E. M. Berti, A. J. S. Salmerón, and F. Benimeli. Human–robot interaction and tracking using low cost 3d vision systems. *Romanian Journal of Technical Sciences - Applied Mechanics*, 7(2):1–15, 2012.

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

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

[6] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. 2009.

[7] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[8] G. Gkioxari, P. Arbeláez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3342–3349. IEEE, 2013.

[9] D. Grest, J. Woetzel, and R. Koch. Nonlinear body pose estimation from depth images. In *Pattern Recognition*, pages 285–292. Springer, 2005.

[10] 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 *3D Vision - 3DV 2013, 2013 International Conference on*, pages 279–286, June 2013.

[11] C. Ionescu, F. Li, and C. Sminchisescu. Latent structured models for human pose estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2220–2227. IEEE, 2011.

[12] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[13] W. Khalil and E. Dombre. *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004.

[14] K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3d scenes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1330–1337. IEEE, 2012.

[15] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 588–595. IEEE, 2013.

[16] 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.

[17] D. R. Faria, C. Premebida, and U. Nunes. A probalistic approach for human everyday activities recognition using body motion from rgb-d images. *IEEE RO-MAN'14: IEEE International Symposium on Robot and Human Interactive Communication*, 2014.

[18] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *Computer Vision–ECCV 2014*, pages 33–47. Springer, 2014.

[19] D. Ramanan. Learning to parse images of articulated bodies. In *Advances in neural information processing systems*, pages 1129–1136, 2006.

[20] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3674–3681. IEEE, 2013.

[21] J. Shan and S. Akella. 3d human action segmentation and recognition using pose kinetic energy. *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 2014.

[22] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, et al. Efficient human pose estimation from single depth images. *Pattern Analy-*

sis and Machine Intelligence, IEEE Transactions on, 35(12):2821–2840, 2013.

[23] 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.

[24] L. Spinello and K. O. Arras. People detection in rgb-d data. In *Intelligent Robots and Systems (IROS), 2011*. IEEE, 2011.

[25] J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from rgbd images. *Plan, Activity, and Intent Recognition*, 64, 2011.

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

[27] C. R. Viala, A. J. S. Salmeron, and E. Martinez-Berti. Calibration of a wide angle stereoscopic system. *OPTICS LETTERS, ISSN 0146-9592, pag 3064-3067.*, 2011.

[28] C. R. Viala, A. J. S. Salmeron, and E. Martinez-Berti. Accurate calibration with highly distorted images. *APPLIED OPTICS, ISSN 0003-6935, pag 89-101*, 2012.

[29] K. Waldron Prof and J. Schmiedeler Prof. *Kinematics*. Springer Berlin Heidelberg, 2008.

[30] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 596–603. IEEE, 2013.

[31] J. Wang, Z. Liu, and Y. Wu. Learning actionlet ensemble for 3d human action recognition. In *Human Action Recognition with Depth Cameras*, pages 11–40. Springer, 2014.

[32] Y. Wang, D. Tran, Z. Liao, and D. Forsyth. Discriminative hierarchical part-based models for human parsing and action recognition. *The Journal of Machine Learning Research*, 13(1):3075–3102, 2012.

[33] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2878–2890, 2013.