published in 2011 IEEE Int. Conf. on Computer Vision workshops, 906-913, Barcelona/Spain

final version ©by IEEE, see IEEE Xplore

On the Effect of Temporal Information on Monocular 3D Human Pose Estimation

Jürgen Brauer*, Wenjuan Gong$^\diamond$, Jordi Gonzàlez$^\diamond$, Michael Arens*
*Fraunhofer IOSB, Ettlingen, Germany
$^\diamond$Computer Vision Center, Universitat Autònoma de Barcelona, Spain
*{juergen.brauer,michael.arens}@iosb.fraunhofer.de, $^\diamond${wenjuan,poal}@cvc.uab.es

Abstract

We address the task of estimating 3D human poses from monocular camera sequences. Many works make use of multiple consecutive frames for the estimation of a 3D pose in a frame. Although such an approach should ease the pose estimation task substantially since multiple consecutive frames allow to solve for 2D projection ambiguities in principle, it has not yet been investigated systematically how much we can improve the 3D pose estimates when using multiple consecutive frames opposed to single frame information.

In this paper we analyze the difference in quality of 3D pose estimates based on different numbers of consecutive frames from which 2D pose estimates are available. We validate the use of temporal information on two major different approaches for human pose estimation – modeling and learning approaches. The results of our experiments show that both learning and modeling approaches benefit from using multiple frames opposed to single frame input but that the benefit is small when the 2D pose estimates show a high quality in terms of precision.

1. Introduction

Estimating the 3D articulation of humans in videos is an important topic in computer vision since the knowledge about the articulation of persons opens the door for behavior analysis based on such 3D pose estimates. If only monocular camera information is available, the task of identifying the 3D pose of persons in videos showing a huge variety of actions, lighting conditions, person occlusions, and cluttered background can be considered as yet unsolved. To ease this problem, the idea of using temporal information for estimating the 3D pose in a frame is obvious since this should allow to solve for ambiguities.

A seminal work that motivated the idea of using several consecutive frames for pose estimation was the work by Johansson [5] with so called Moving Light Displays (MLD), which are a small number of light sources attached to the body of a person in a dark scene. Johansson showed that a small number of points of the human body is sufficient to recognize and discriminate human poses and motions correctly if sequences of these points are presented. Rashid [7] showed that such sequences of 2D points allow to identify the body parts of two walking persons even in the case of a short overlap of both 2D point sets.

One popular way to incorporate temporal information is to improve the quality of pose estimation with motion models. For example, Urtasun *et al*. [14] use learned motion models for human pose tracking. The temporal correlations between consecutive frames are incorporated into motion models. Human pose estimation is then confined by this learned motion models. Instead of learning motion models, the methods presented here incorporate temporal information by using several consecutive frames as input. In this way, we can get rid of the learning phase for motion models.

There is a huge variety in how multiple consecutive frames are used for human pose estimation. Singh and Nevatia [11] *e.g.* track individual body parts over multiple frames using a particle filtering approach that incorporates kinematic constraints. Andriluka *et al*. [1] first identify complete 2D poses for single frames and then uses sequences of 2D poses over multiple frames ('2D tracklets') as input for a 3D pose estimator. Daubney *et al*. [3] *e.g.* use as observational data a sparse cloud of features extracted using the Kanade-Lucas-Tomasi (KLT) feature tracker. Using the motion over multiple frames of such features, low-level part detectors are learnt directly from motion capture data.

Interestingly, it has not yet been investigated systematically how the number of consecutive frames influence 3D human pose estimation results. Intuitively, we would say using more frames will be better, but it is unclear how much we gain regarding the quality of 3D pose estimates compared to single frame based 3D pose estimation. Further, we do not know whether there is some input window size where we run into a saturation of the 3D pose estimation performance gain and simultaneously waist more and more

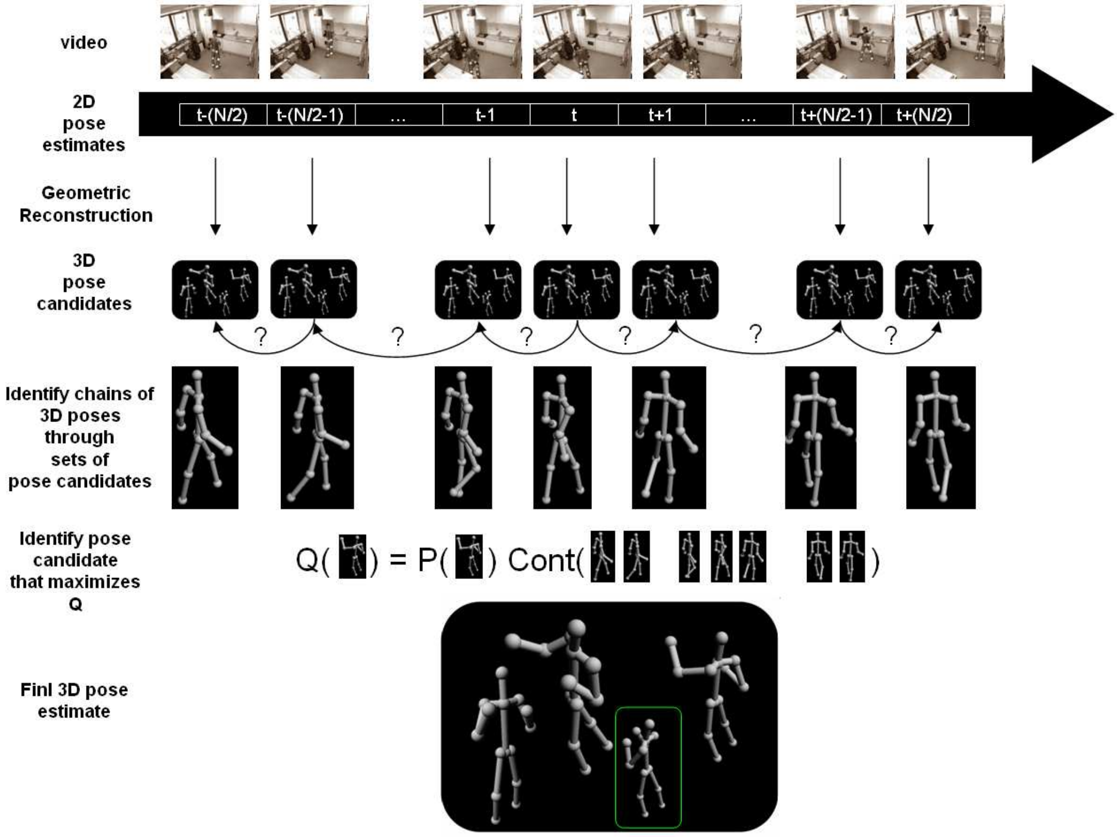

Figure 1. Geometric reconstruction of 3D poses. To integrate information from multiple consecutive frames into the 3D pose estimation for one frame at time t we first reconstruct all 3D pose candidates for each time step independently and then weight the probability $P()$ of each pose candidate at time t by a measure of continuity $Cont()$ which measures how good can we find a sequence of poses through previous time steps, time step t and successive time steps such that the joint angles only change continuously.

computing time by processing too big input window sizes. It is further unclear whether there exists something like an ideal window size at all. The contribution of this paper is to present a detailed evaluation of the 3D pose reconstruction performance for different scenarios (actions, viewpoints, persons, datasets) as a function of the input window size (see section 3) and thereby to give answers to these open questions.

Since there are two main classes of 3D pose estimation approaches – learning and modeling based – and it is not feasible to perform the evaluation for all approach variants we choose the most typical representative of each class of approaches for the evaluation. Learning approaches try to learn a mapping from images to 3D poses using training examples and adapt some mapping using *e.g.* support vector regressors, relevance vector regressors, or Gaussian process regressors. Modeling approaches try to model this mapping from 2D to 3D poses explicitly by using knowledge about the inverse of the 3D to 2D mapping.

For the class of modeling approaches we choose a geometric reconstruction approach – originally presented for a restricted parallel projection camera model [12], used in several following works (*e.g.* [4], [6]), and recently extended to a realistic perspective projection camera model [2]. In section 2.1 we present a detailed explanation of this geometric method. For the class of learning approaches we choose the Gaussian Process regression since it is successfully used in many pose estimation works (*e.g.* [9], [15]). Refer to section 2.2 for an explanation of this method.

2. Usage of Temporal Information for 3D Pose Estimation

In this section, we explain how to incorporate temporal information into pose estimation by taking several consecutive frames as input. We use two different methods for validation. geometric reconstruction method (in section 2.1) and regression method (in section 2.2).

2.1. Geometric Reconstruction Method

Taylor's [12] work is the most commonly used approach within the class of modeling approaches. It assumes a scaled orthographic projection camera model, which means that a 3D object point (x, y, z) is mapped to its corresponding 2D image point (u, v) by $u = s \cdot x, v = s \cdot y$, i.e. 3D objects points are assumed to be projected to the 2D image by a parallel projection with a subsequent scaling with scaling factor s. The basic principle in this geometric reconstruction method is to use the foreshortening information of projected limb lengths between two projected points (u_1, v_1) and (u_2, v_2) and compare these with the known 3D limb lengths l between two body marker points (x_1, y_1, z_1) and (x_2, y_2, z_2) of the 3D boy model. This allows to reconstruct the displacement $\Delta_z := z_1 - z_2$ of the limb in z direction based on the measured length of the foreshortened limb in the 2D image by

$$\Leftrightarrow \Delta_z = \pm\sqrt{l^2 - \frac{(u_1 - u_2)^2 + (v_1 - v_2)^2}{s^2}} \tag{1}$$

The displacement Δ_z can be reconstructed for one limb only up to a sign ($+$ or $-$) ambiguity in equation 1 since we cannot decide which of the limb endpoints (x_1, y_1, z_1), (x_2, y_2, z_2) is nearer to the camera. For a body model with N limbs, these two reconstruction possibilities for one limb mean 2^N reconstruction possibilities for the whole body pose. To solve this ambiguity automatically different approaches have been suggested [4, 6]. Other works [17] tackle the problem of how to determine the unknown scale factor s by augmenting the set of projection constraints (Equation 1 by further constraints. Nevertheless, the main problem of the geometric reconstruction method is the unrealistic camera model. The model assumes that the projected size of a person or a limb does not depend on its distance to the camera which is not true for real cameras. Note that the z coordinate has no influence on the resulting (u, v) coordinate. In [2] it was recently shown how to augment the geometric reconstruction method to perspective projections. The perspective camera models the projection of a 3D point (x, y, z) to a 2D point (u, v) by $u = f\frac{x}{z} + c_0$, $v = f\frac{y}{z} + c_1$. f is called focal length, (c_0, c_1) is called principal point. In [2] it was shown that there are two solutions for the z_i coordinate of a child marker given an already reconstructed z_j coordinate of a parent marker:

$$z_{i_{1/2}} = -\frac{Cz_j}{2A} \pm \sqrt{(\frac{Cz_j}{2A})^2 - (\frac{B}{A}z_j^2 - \frac{f^2 l_{ij}^2}{A})} \tag{2}$$

with $A = u_i^2 + v_i^2 + f^2$, $B = u_j^2 + v_j^2 + f^2$, and $C = 2(u_i u_j + v_i v_j + f^2)$.

This geometric reconstruction algorithm for 3D poses assumes that we provide limb lengths l_{ij} (connecting marker i with j), and the 2D coordinates (u, v) of the markers within the image. The focal length and principal point is supposed to be provided by camera calibration, the limb lengths can be taken from a person of average size. The 2D coordinates are supposed to be provided by a 2D pose estimator.

Comparing equation 1 with equation 2 shows an important difference. For the case of an orthographic projection camera model, the z coordinate could be computed in the original geometric reconstruction approach independently for each limb. In contrast, for the perspective projection camera model, we first have to start with an estimate for the z coordinate of the root marker of the kinematic tree, then we can apply equation 1 in a recursive manner: having computed the z coordinate for a parent marker, we can compute the two possible solutions for the z coordinate of the child marker and step down further in the kinematic tree. Since there are still two solutions for the z coordinate (either $+$ and $-$ in equation 2) we end up with a binary reconstruction tree with 2^N mathematically possible poses. Following the approach presented in in [2] we reduce this huge number of pose candidates already during binary reconstruction tree traversal by checking for abnormal joint angles based on anatomical joint limits in the knees and elbows and prune branches of the reconstruction tree whenever we encounter anatomical violations. A final 3D pose estimate is selected by assigning a probability

$$P(\vec{p}) = \prod P(\vec{j_i}) \tag{3}$$

to each pose candidate $\vec{p}$, where $P(\vec{j_i} = (\alpha, \beta, \gamma))$ is the probability to find a joint in a certain configuration $\vec{j_i} = (\alpha, \beta, \gamma)$ (the three Euler angles) which can be learned by observing motion capture sample data. The z coordinate of the root marker can be estimated by the distance of the person to the image plane. In [2] the proposed solution for the estimation of the person to camera distance was reconstructing all possible poses using different distance estimates and then choose the distance where the average pose probability takes on a maximum. This approach is successful for estimating the person ↔ camera distance since for distances different from the ground truth distance, the reconstructed poses have to be squeezed (distance too small) or pulled apart (distance estimate too big) into the perspectives rays bundle which in turn results in unlikely joint angles and small pose probabilities. For further details we refer the reader to [2].

The method described so far uses only input from a single frame: given a 2D input pose estimate for the current frame, we reconstruct its corresponding 3D pose. We now extend this approach to inputs from multiple consecutive frames. For this we introduce a new definition for the probability of a pose candidate which will still depend on the product of the individual joint angle probabilities but additionally depend on how good we can find a sequence of previous and successive 3D poses such that the joint angles change continuously.

More exactly, we want to estimate the 3D pose for frame t given an input window of size N centered at this frame, *i.e.* we assume we have 2D pose estimates for frames $t-(N/2)$,

$t-(N/2-1)$, ..., $t-1$, t, $t+1$, ..., $t+(N/2-1)$, $t+(N/2)$. There is no straightforward extension for the geometric reconstruction as in the case for regression approaches, where we simply extend the input vector (containing then 2D pose estimates from multiple frames) and learn the mapping to the output vector (3D pose estimate) by function regression. Here we propose to reconstruct the set of all 3D pose estimates for each individual time step at first and then try to find a plausible sequence of 3D pose estimates for the N frames. For this we first define a chain of 3D pose estimates for each pose candidate $\vec{p}_t$ at the current time step t by searching for each of these pose candidates first, the most similar 3D pose candidates $\vec{p}_{t-1}$ at time step $t-1$ and $\vec{p}_{t+1}$ at time step $t+1$. Then we continue by searching for the most similar pose candidate $\vec{p}_{t-2}$ to $\vec{p}_{t-1}$ at time step $t-2$ and the most similar pose candidate $\vec{p}_{t+2}$ to $\vec{p}_{t+1}$ at time step $t+2$, etc. We end up with a chain $C=(\vec{p}_{t-(N/2)},...,\vec{p}_{t+(N/2)})$ of 3D pose estimates for each pose candidate $\vec{p}_t$ at time step t. For exploiting time information, we will extend the pose probability definition for single frame input (Equation 3) which measures the probability of a pose based on the joint angle probabilities by some measure $Cont(C)$ of the continuity of its associated chain of poses:

$$Q(\vec{p}_t) = P(\vec{p}_t) \cdot Cont(C) \tag{4}$$

where

$$Cont(C) = 1/(\sum_{i=-N/2}^{i=N/2-1} |\vec{p}_{t+i} - \vec{p}_{t+i+1}|) \tag{5}$$

i.e. $Cont(C)$ is defined such that it measures the difference between successive poses $\vec{p}_{t-(N/2)}$ and $\vec{p}_{t-(N/2-1)}$, $\vec{p}_{t-(N/2-1)}$ and $\vec{p}_{t-(N/2-2)}$, etc. and fosters chain of poses (by computing the reciprocal) such that the joint angles change smoothly. Thereby we assume that for a plausible 3D pose candidate we can find previous and successive 3D poses such that we can stick them together to a sequence of poses 'through' $\vec{p}_t$. At a first glance this measure $Cont(C)$ seems to prefer sequences of constant poses but since for each time step there are only 3D pose candidates that are consistent with the corresponding 2D input poses that will change continuously this is not true in practice.

The experiments described in section 3 were conducted with this time information integrating probability measure Q.

2.2. Regression Method

Boosting regression method using temporal information is quite straightforward. The main procedure is shown in figure 2. The upper row shows pose estimation with regression method and the lower row shows pose estimation with regression method with temporal information incorporated. The main idea is to concatenate features from several consecutive frames instead of one as input.

From the detected body part positions of a performer we take 13 body parts. The 2D body part positions are collected within a 26-dimensional vector BP:

$$BP = [x_1, y_1, x_2, y_2, \ldots, x_i, y_i, \ldots, x_{12}, y_{12}, x_{13}, y_{13}] \tag{6}$$

where (x_i, y_i) is the 2D position of the i-th body part. For representing the 2D pose independently of the persons's size and distance to the camera, we normalize this 2D pose vector:

$$BP_{norm} = (BP + M_{off}) * M_{scale} \tag{7}$$

where $*$ means element-wise multiplication and

$$\begin{aligned} M_{scale} &= [\frac{1}{y_{range}}, \frac{1}{y_{range}}, \ldots, \frac{1}{y_{range}}, \frac{1}{y_{range}}] && (8)\\ M_{off} &= [x_{off}, y_{off}, x_{off}, y_{off}, \ldots, x_{off}, y_{off}] && (9)\\ x_{off} &= -min(X) + (y_{range} - x_{range})/2 && (10)\\ y_{off} &= -min(Y) && (11) \end{aligned}$$

For upright standing persons, the range of y coordinate values is typically bigger compared to the range of x coordinate values. For this, we normalize both x and y coordinates by y range in each frame (M_{scale}). This makes sure, that we keep the aspect ratio of the performer and that normalized y coordinates range from 0 to 1.

To use temporal information we concatenate features from several consecutive frames. For example, if we take consecutive frames $t-1$, t and $t+1$ for the estimation of the 3D pose in frame t, then the input vector for the regression method at time t is represented as:

$$F_t = [BP_{norm}^{t-1}, BP_{norm}^{t}, BP_{norm}^{t+1}], \tag{12}$$

where BP_{norm}^t represents the normalized 2D pose at frame t.

For 3D pose representation, we model a human pose using twelve rigid body parts: hip, torso, shoulder, neck, two thighs, two lower legs, two upper arms and two forearms. The pose of an actor in an image frame is represented as a vector of direction cosines, i.e. the cosines of the angles between the limb direction vectors and the three coordinate axes of the root coordinate system. The overall posture of the subject for a frame is represented using a vector of direction cosines measured on twelve limbs. We use the same representations for human posture as in [8]. This results in a 36-dimensional representation of the pose:

$$\psi = [\cos\theta_1^x, \cos\theta_1^y, \cos\theta_1^z, \ldots, \cos\theta_{12}^x, \cos\theta_{12}^y, \cos\theta_{12}^z], \tag{13}$$

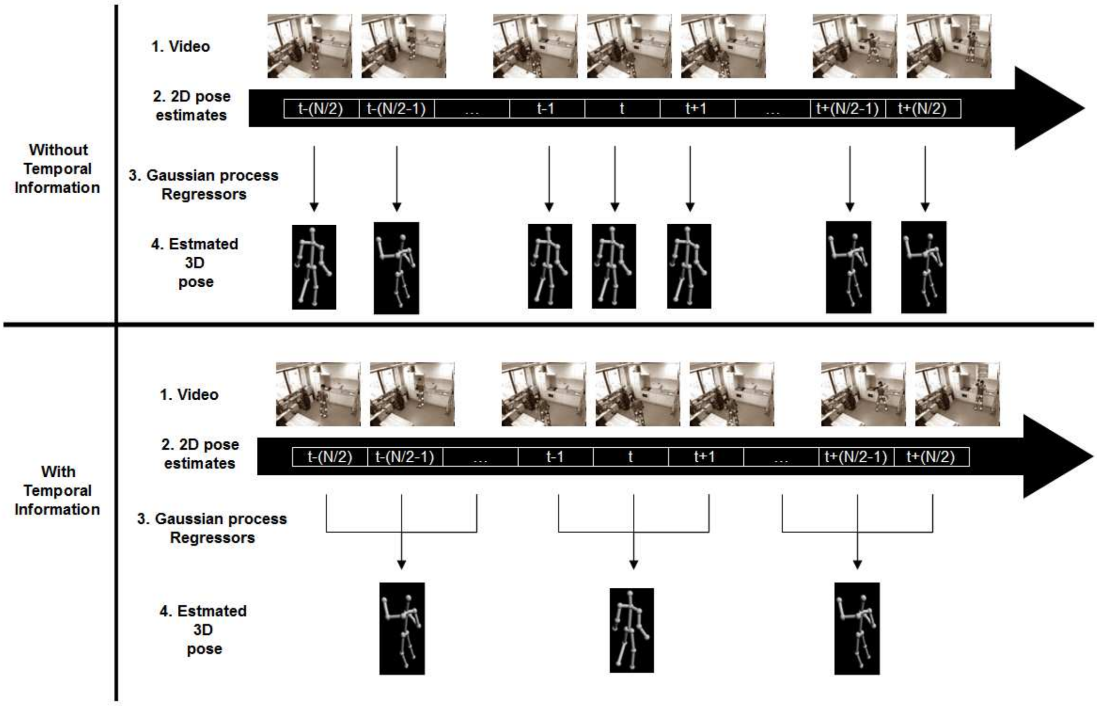

Figure 2. Gaussian Process regression of 3D poses using input from single or multiple consecutive frames. To make use of temporal information we concatenate the 2D input vectors from several frames into one large input vector that will be mapped to a 3D pose by Gaussian Process regression.

where θ_l^x, θ_l^y and θ_l^z are the angles between the limb l and the axes of the root coordinate system in the hip.

With 2D body part positions and corresponding 3D pose representation, we can train a set of Gaussian processors. The main idea of Gaussian process regression is to map unknown test data to a prediction by interpolating the training data weighted by the correlation between the training and test data. Given a 2D pose estimate which is represented as the $26 \times (number-of-consecutive-frames)$ dimensional vector, we train one Gaussian process to predict each of the 36 dimensions of the 3D pose vector ψ separately. For the Gaussian process training and prediction we used a reference implementation[1].

3. Experiments

In this section, we describe the set of experiments we conducted to analyze the 3D pose estimation performance of the geometric reconstruction and Gaussian process regression method as a function of different input window sizes. We further give details about the 3D pose error measure used.

3.1. Experiment definitions

Both the public available HumanEva [10] and the TUM kitchen dataset [13] are suitable for evaluations concerning 3D pose estimation quality since both datasets provide 3D motion capture ground truth data. This allows to compute an error for each estimated pose by comparing it with its corresponding ground truth frame. Furthermore, intrinsic and extrinsic camera parameters are provided for both datasets as well. Thereby we can project the 3D poses into the image and generate 2D ground truth poses as well. The datasets contain sequences where different subjects (4 for HumanEva, 4 for TUM kitchen) perform different actions (walking, boxing, laying a kitchen table, etc.) recorded from different viewpoints (7 for HumanEva, 4 four TUM kitchen). We use 4 categories of experiments:

1. *train* on a sequence recorded from one camera view → *test* on a sequence recorded from another view (1a/1b/1c/1d). The change of viewpoint can be weak (1a/1b) or strong (1c/1d).

2. *train* on a sequence comprising a subject S_i → *test* on a sequence comprising another subject S_j [2] (2a/2b),

3. *train* on a sequence showing one action class A_1 → *test* on a sequence showing another action class A_2

[1] http://www.gaussianprocess.org/gpml/code/matlab/doc/

[2] Person S_i within the TUM kitchen dataset is different from the person S_i within the HumanEva dataset

Exp.	training	testing	change of
1a	TUM, 0-0-cam3, S1	TUM-0-0-cam2, S1	viewpoint (weak)
1b	HE, walk-cam1, S1	HE, walk-cam2, S1	viewpoint (weak)
1c	TUM, 0-0-cam1, S1	TUM-0-2-cam3, S1	viewpoint (strong)
1d	HE, box-cam1, S1	HE, box-cam2, S1	viewpoint (strong)
2a	TUM, 0-0-cam3, S1	TUM-0-3-cam3, S2	person
2b	HE, walk-cam1, S1	HE, walk-cam1, S2	person
3a	HE, walk-cam1, S2	HE, box-cam1, S2	action
3b	HE, box-cam1, S2	HE, walk-cam1, S2	action
4a	HE, walk-cam2, S1	TUM, 0-2-cam3, S2	dataset
4b	TUM, 0-2-cam3, S2	HE, walk-cam2, S1	dataset

Table 1. Experiments definition.

Exp.	**Geometric reconstruction error [°]**						**Regression error [°]**					
	Nr of Frames						**Nr of frames**					
	1	3	5	9	17	33	1	3	5	9	17	33
1a	6.12	5.32	**5.31**	**5.31**	5.33	5.36	**0.12**	5.32	5.26	5.39	5.55	5.66
1b	7.47	7.44	7.43	7.43	**7.42**	7.43	1.39	1.25	1.20	**1.19**	1.21	1.20
1c	**5.24**	5.33	5.35	5.34	5.36	5.40	5.48	**5.41**	5.75	5.62	5.62	5.60
1d	8.15	**8.10**	8.12	8.13	8.14	8.12	4.49	4.55	4.63	4.47	**4.13**	4.25
2a	**6.53**	7.65	7.65	7.69	7.69	7.78	5.52	4.97	**4.72**	4.96	5.08	5.40
2b	8.61	8.01	8.01	8.01	7.91	**7.78**	3.87	3.49	**3.44**	3.70	3.64	3.94
3a	16.65	**15.40**	15.43	15.49	15.60	15.66	11.48	11.54	**11.00**	11.39	11.36	11.55
3b	9.10	9.37	8.68	8.66	8.51	**8.31**	9.34	**8.66**	8.96	9.64	9.53	10.15
4a	7.57	7.38	7.35	7.32	7.35	**7.31**	8.40	8.29	8.33	8.33	8.48	**8.28**
4b	8.07	**7.87**	7.88	**7.87**	**7.87**	7.89	7.08	6.85	**6.30**	6.48	6.68	6.79

Table 2. Experimental results. We present the 3D pose reconstruction error for each experiment using different input window sizes. The best 3D pose reconstruction performance for each experiment and each method is marked in bold.

(3a/3b),

4. *train* on a sequence from HumanEva (TUM kitchen) dataset → *test* on a sequence from the other dataset, i.e. TUM kitchen (HumanEva) (4a/4b)

3.2. Error measurement

Both approaches, the regression and the geometric reconstruction method, map 2D input poses to 3D pose estimates. We test on ground truth 2D input poses with different numbers of frames as input. In the experiments, we exponentially select 1, 3,5, 9, 17, 33 and 65 as window sizes. Experiment definitions are shown in Table 1, while the results are shown in Table 2.

Since we use the same input 2D poses for both experiments this allows us to compare both approaches - the regression and the geometric reconstruction - based on their 3D pose estimation performance. The performance is measured by the average angular error of the estimated 3D poses compared to the ground truth 3D poses. If predicted limb angles $\hat{\Theta}$ and ground truth limb angles Θ are denoted as

$$\begin{aligned}\hat{\Theta} &= [\hat{\theta}^x_{l_1}, \hat{\theta}^y_{l_1}, \hat{\theta}^z_{l_1}, \ldots, \hat{\theta}^x_{l_{14}}, \hat{\theta}^y_{l_{14}}, \hat{\theta}^z_{l_{14}}] \qquad (14)\\ \Theta &= [\theta^x_{l_1}, \theta^y_{l_1}, \theta^z_{l_1}, \ldots, \theta^x_{l_{14}}, \theta^y_{l_{14}}, \theta^z_{l_{14}}] \qquad (15)\end{aligned}$$

then the average angular error is defined as:

$$Err_{Ang} = \frac{\sum_{i=1}^{J} |\Theta_i - \hat{\Theta}_i| \bmod 180^\circ}{J} \qquad (16)$$

where $J = 3 \cdot 14$ (3 Euler angles, 14 limbs).

We also introduce joint position error measurement for visualizing the effect to single joint after incorporating temporal information. If we denote these estimated marker positions $\hat{\mathbf{P}}$ and ground marker positions $\mathbf{P}$:

$$\begin{aligned}\hat{\mathbf{P}} &= [\hat{x}_1, \hat{y}_1, \hat{z}_1, \ldots, \hat{x}_{15}, \hat{y}_{15}, \hat{z}_{15}] \qquad (17)\\ \mathbf{P} &= [x_1, y_1, z_1, \ldots, x_{15}, y_{15}, z_{15}], \qquad (18)\end{aligned}$$

then the average marker position error is defined as

$$Err_{pos} = \frac{\sum_{i=1}^{M} |\mathbf{P}_i - \hat{\mathbf{P}}_i|}{M}. \qquad (19)$$

where $M = 3 \cdot 15$ (x/y/z coordinates, 15 markers). Err_{Ang} is specified in degrees, Err_{pos} in mm.

3.3. Results

The results of our experiments are shown for both methods – the geometric reconstruction of 3D poses and the Gaussian process regression of 3D poses – in Table 2. Note

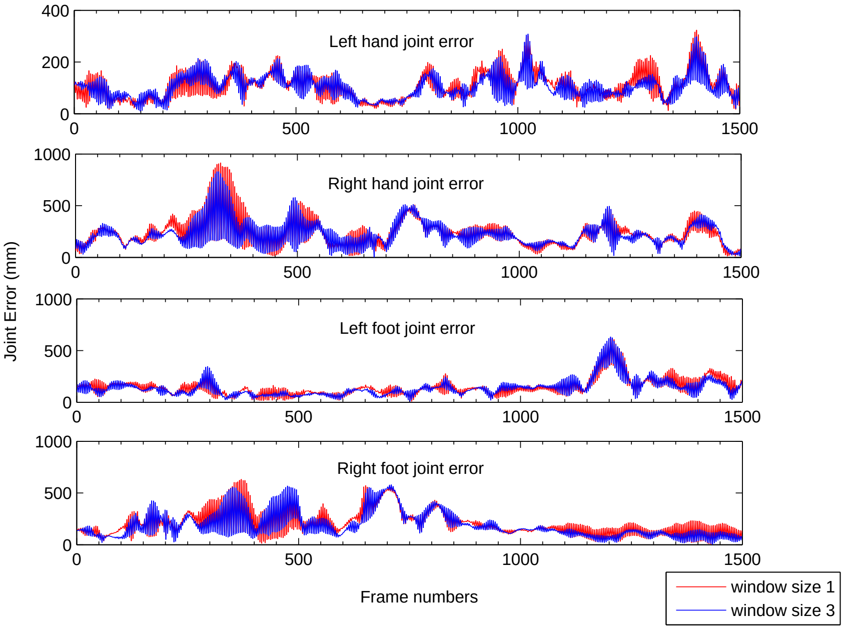

Figure 3. Visualized joint position errors of experiment 2a with regression method. We compared errors from two window size: window size 1 (red) and window size 3 (blue). Four joints are considered: left hand, right hand, left foot and right foot. Error measurement is defined in section 3.2, equation 19.

the bold numbers which highlight the smallest 3D pose error for each experiment and window size. Considering these results we can see a general trend: incorporating temporal information improves the pose estimation results in the sense that the 3D pose error can be reduced, for the geometric reconstruction method in all experiments but 1c,2a and for the regression method in all experiments but 1a. In some experiments as e.g. for the geometric reconstruction in experiment 2b and 4a using more and more 2D input poses leads indeed to a continuous decrease of the 3D pose error.

Nevertheless, our expectations 1.) to find a clear optimum window size for each method and 2.) to find a strong improvement concerning the 3D pose estimation performance when using 2D input poses from multiple frames were not confirmed. Note that although in most experiments the error decreased, the reduction is rather small. Since these expectations were not fulfilled independently for both methods, we think it is probably not because of a special or wrong way in which we incorporate the temporal information in the geometric reconstruction approach although there are of course a lot of other possibilities to incorporate the additional 2D input poses into the geometric reconstruction approach. For the regression approach the concatenation of the individual 2D input poses to one big input vector is a quite common strategy.

A possible explanation for our results is that we worked with 2D ground truth poses as input. This could mean that the benefit of using input poses from additional frames is rather marginal since a perfect 2D pose estimate for the current frame could already be sufficient for the estimation of the 3D pose. This in turn would mean that the better the 2D pose estimates are in average, the less there is the need to use additional input frames and thereby saving computing time.

Figure 3 shows exemplary marker estimation position errors of left hand, right hand, left foot and right foot retrieved from final 3D pose estimates. These four body markers are more representative as e.g. the head since they show much more articulation variety. From the figure, we can observe that at some frames, *e.g.* frame 1250 to frame 1300, incorporating temporal information helps enhance pose estimation accuracy dramatically. While for some frames, *e.g.* frame 400 to frame 500, temporal information improves estimation performance for left foot joint while decreases performance for right foot joint. Our explanation for this is: left foot joint moves smoother than right foot joint in these frames which results the benefit from temporal information.

4. Conclusion and Future Work

In this paper, we explored the possibility of incorporating temporal information to improve 3D pose estimation performance. We tested two different methods, each representative for a whole set of approaches: the geometric reconstruction as the most commonly used modeling approach and the Gaussian process regression as it is now the most widespread regression method in the 3D pose estimation literature. For the geometric method, we showed how to incorporate temporal information by augmenting the single frame probability measure based on joint angle probabilities by a measure of continuity of joint angle changes. For the regression method, we integrated temporal information by taking several consecutive frames and concatenating the corresponding 2D poses to one input vector.

The results showed an advantage of using input poses from multiple consecutive frames over the single frame input situation. Though the benefit of the temporal input was not as big as expected which we trace back to the fact that perfect 2D input poses were used. The preliminary conclusion is that the window size should probably be adapted according to the quality of the 2D pose estimates such that it is bigger if the quality of 2D pose estimates is low.

For our experiments we used two different challenging datasets but the ground truth 2D pose was always available even for frames in which in some camera views some parts of the person were occluded (e.g. person standing behind the kitchen table). If the 2D poses are provided by a 2D pose estimator which only uses such monocular camera image information where occlusions are present, the 2D pose estimates will be not such perfect any longer and the results obtained will probably differ from the results obtained here. An interesting dataset in this context is the SCOVIS [16] dataset, since it contains many occlusion situations. We then expect a significantly bigger gain of using temporal information.

Future work will try to quantify the 3D pose reconstruction performance as a function of the window size for the case of noisy 2D input poses. Further, we will try out alternatives for the use of temporal information in the geometric reconstruction and regression approach, e.g. by an update approach where all 2D pose estimates from the beginning of the video up to the current frame are integrated and compare its effect on the 3D pose estimation performance with results of the methods for incorporating time information presented here.

Acknowledgments

This work is supported by the Spanish projects TIN2009-14501-C02-02, Consolider MIPRCV (CSD200700018), and Avanza I+D ViCoMo (TSI-020400-2009-133).

References

[1] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *Proc. of CVPR 2010*, USA, 2010. 1

[2] J. Brauer and M. Arens. Reconstructing the missing dimension: From 2d to 3d human pose estimation. In *Proc. of REACTS 2011 - REcognition and ACTion for Scene Understanding - in conj. with CAIP 2011, 14th Intern. Conf. on Computer Analysis of Images and Patterns*, pages 25–39, Spain, Málaga, 2011. 2, 3

[3] B. Daubney, D. Gibson, and N. Campbell. Monocular 3d human pose estimation using sparse motion features. In *IEEE worshop on Tracking Humans for Evaluation of their Motion in Image Sequences 2009 - Held in conjunction with ICCV*, October 2009. 1

[4] H. Jiang. 3d human pose reconstruction using millions of exemplars. In *Proc. of 20th ICPR*, pages 1674–1677, 2010. 2, 3

[5] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14:201–211, 1973. 1

[6] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *PAMI*, 28(7):1052–1062, 2006. 2, 3

[7] R. Rashid. Towards a system for the interpretation of moving light display. *PAMI*, 2(6):574–581, 1980. 1

[8] I. Rius, J. Gonzàlez, J. Varona, and F. X. Roca. Action-specific motion prior for efficient bayesian 3d human body tracking. *Pattern Recognition*, 42(11):2907–2921, 2009. 4

[9] M. Salzmann and R. Urtasun. Implicitly constrained gaussian process regression for monocular non-rigid pose estimation. *In Proc. of NIPS, 2010*, pages 2065–2073, 2010. 2

[10] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical report, Brown University, 2006. 5

[11] V. K. Singh and R. Nevatia. Monocular human pose tracking using multi frame part dynamics. In *Proceedings of the 2009 international conference on Motion and video computing*, WMVC'09, pages 1–8, Washington, DC, USA, 2009. IEEE Computer Society. 1

[12] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *CVIU*, 80:349–363, 2000. 2

[13] M. Tenorth, J. Bandouch, and M. Beetz. The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In *THEMIS workshop. In conj. with ICCV 2009*. 5

[14] R. Urtasun, D. J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3d human body tracking. *CVIU*, 104:157–177, November 2006. 1

[15] R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. *IEEE Int. Conf. on Computer Vision*, 1:403–410, 2005. 2

[16] A. Voulodimos, D. Kosmopoulos, G. Vasileiou, E. S. Sardis, A. D. Doulamis, V. Anagnostopoulos, and T. Varvarigou. A dataset for workflow recognition in industrial scenes. In *2011 IEEE Int. Conf. on Image Processing*, September 2011. 8

[17] X. K. Wei and J. Chai. Modeling 3d human poses from uncalibrated monocular images. In *IEEE 12th Intern. Conf. on Computer Vision*, pages 1873–1880, October 2009. 3