A Dual-Source Approach for 3D Pose Estimation from a Single Image

Hashim Yasin\textsuperscript{*1}, Umar Iqbal\textsuperscript{*2}, Björn Krüger\textsuperscript{3}, Andreas Weber\textsuperscript{1}, and Juergen Gall\textsuperscript{2}

\textsuperscript{1}Multimedia, Simulation, Virtual Reality Group, University of Bonn, Germany
\textsuperscript{2}Computer Vision Group, University of Bonn, Germany
\textsuperscript{3}Gokhale Method Institute, Stanford, USA

Abstract

One major challenge for 3D pose estimation from a single RGB image is the acquisition of sufficient training data. In particular, collecting large amounts of training data that contain unconstrained images and are annotated with accurate 3D poses is infeasible. We therefore propose to use two independent training sources. The first source consists of images with annotated 2D poses and the second source consists of accurate 3D motion capture data. To integrate both sources, we propose a dual-source approach that combines 2D pose estimation with efficient and robust 3D pose retrieval. In our experiments, we show that our approach achieves state-of-the-art results and is even competitive when the skeleton structure of the two sources differ substantially.

1. Introduction

Human 3D pose estimation from a single RGB image is a very challenging task. One approach to solve this task is to collect training data, where each image is annotated with the 3D pose. A regression model, for instance, can then be learned to predict the 3D pose from the image \cite{6, 14, 12, 2, 7, 16, 17}. In contrast to 2D pose estimation, however, acquiring accurate 3D poses for an image is very elaborate. Popular datasets like HumanEva \cite{23} or Human3.6M \cite{13} synchronized cameras with a commercial marker-based system to obtain 3D poses for images. This requires a very expensive hardware setup and the requirements for marker-based systems like studio environment and attached markers prevent the capturing of realistic images.

Instead of training a model on pairs consisting of an image and a 3D pose, we propose an approach that is able to incorporate 2D and 3D information from two different training sources. The first source consists of images with annotated 2D pose. Since 2D poses in images can be manually annotated, they do not impose any constraints regarding the environment from where the images are taken. Indeed any image from the Internet can be annotated and used. The second source is accurate 3D motion capture data captured in a lab, e.g., as in the CMU motion capture dataset \cite{8} or the Human3.6M dataset \cite{13}. We consider both sources as independent, i.e., we do not know the 3D pose for any training image. To integrate both sources, we propose a dual-source approach as illustrated in Fig. 1. To this end, we first convert the motion capture data into a normalized 2D pose space, and separately learn a regressor for 2D pose estimation from the image data. During inference, we estimate the 2D pose and retrieve the nearest 3D poses using an approach that is robust to 2D pose estimation errors. We then jointly estimate a mapping from the 3D pose space to the image, identify wrongly estimated 2D joints, and estimate the 3D pose. During this process, the 2D pose can also be refined and the approach can be iterated to update the estimated 3D and 2D pose. We evaluate our approach on two popular datasets for 3D pose estimation. On both datasets, our approach achieves state-of-the-art results and we provide a thorough evaluation of the approach. In particular, we analyze the impact of differences of the skeleton structure between the two training sources, the impact of the accuracy of the used 2D pose estimator, and the impact of the similarity of the training and test poses.

2. Related Work

A common approach for 3D human pose estimation is to utilize multiple images captured by synchronized cameras \cite{5, 24, 32}. The requirement of a multi-camera system in a controlled environment, however, limits the applicability of these methods. Since 3D human pose estimation from a single image is very difficult due to missing depth information, depth cameras have been utilized for human pose estimation \cite{4, 22, 11}. However, current depth sensors are limited to indoor environments and cannot be used in unconstrained scenarios. Earlier approaches for monocular 3D human pose estimation \cite{6, 1, 27, 2, 7, 18} utilize discriminative methods to learn a mapping from local image features (e.g. HOG, SIFT, etc.) to 3D human pose or use a CNN \cite{16, 17}. Since local features are sensitive to noise, these methods often assume that the location and scale of the human is given, e.g., in the form of an accurate bounding
Figure 1: Overview. Our approach relies on two training sources. The first source is a motion capture database that contains only 3D poses. The second source is an image database with annotated 2D poses. The motion capture data is processed by pose normalization and projecting the poses to 2D using several virtual cameras. This gives many 3D-2D pairs where the 2D poses serve as features. The image data is used to learn a pictorial structure model (PSM) for 2D pose estimation where the unaries are learned by a random forest. Given a test image, the PSM predicts the 2D pose which is then used to retrieve the normalized nearest 3D poses. The final 3D pose is then estimated by minimizing the projection error under the constraint that the solution is close to the retrieved poses, which are weighted by the unaries of the PSM. The steps (red arrows) in the dashed box can be iterated by updating the binaries of the PSM using the retrieved poses and updating the 2D pose.

While the approach [12] still relies on the known silhouette of the human body, it partially overcomes the limitations of local image features by segmenting the body parts and using a second order hierarchical pooling process to obtain robust descriptors. Instead of predicting poses with a low 3D joint localization error, an approach for retrieving semantic meaningful poses is proposed in [19].

The 3D pictorial structure model (PSM) proposed in [14] combines generative and discriminative methods. Regression forests are trained to estimate the probabilities of 3D joint locations and the final 3D pose is inferred by the PSM. Since inference is performed in 3D, the bounding volume of the 3D pose space needs to be known and the inference requires a few minutes per frame.

Besides of a-priori knowledge about bounding volumes, bounding boxes or silhouettes, these approaches require sufficient training images with annotated 3D poses. Since such training data is very difficult to acquire, we propose a dual-source approach that does not require training images with 3D annotations, but exploits existing motion capture datasets to estimate the 3D human pose.

Estimating 3D human pose from a given 2D pose by exploiting motion capture data has been addressed in a few works [26, 21, 33, 25, 29]. In [33], the 2D pose is manually annotated in the first frame and tracked in a video. A nearest neighbor search is then performed to retrieve the closest 3D poses. In [21] a sparse representation of 3D human pose is constructed from a MoCap dataset and fitted to manually annotated 2D joint locations. The approach has been extended in [29] to handle poses from an off-the-shelf 2D pose estimator [31]. The same 2D pose estimator is also used in [26, 25] to constrain the search space of 3D poses. In [26] an evolutionary algorithm is used to sample poses from the pose space that correspond to the estimated 2D joint positions. This set is then exhaustively evaluated according to some anthropometric constraints. The approach is extended in [25] such that the 2D pose estimation and 3D pose estimation are iterated. In contrast to [21, 29, 26], [25] deals with 2D pose estimation errors. Our approach also estimates 2D and 3D pose but it is faster and more accurate than the sampling based approach [25].

Action specific priors learned from the MoCap data have also been proposed for 3D pose tracking [28, 3]. These approaches, however, are more constrained by assuming that the type of motion is known in advance.

3. Overview

In this work, we aim to predict the 3D pose from an RGB image. Since acquiring 3D pose data in natural environments is impractical and annotating 2D images with 3D pose data is infeasible, we do not assume that our training data consists of images annotated with 3D pose. Instead, we propose an approach that utilizes two independent sources of training data. The first source consists of motion capture data, which is publically available in large quantities and that can be recorded in controlled environments. The second source consists of images with annotated 2D poses, which is also available and can be easily provided by humans. Since we do not assume that we know any relations between the sources except that the motion capture
data includes the poses we are interested in, we preprocess
the sources first independently as illustrated in Fig. 1. From
the image data, we learn a pictorial structure model (PSM)
to predict 2D poses from images. This will be discussed in
Section 4. The motion capture data is prepared to efficiently
retrieve 3D poses that could correspond to a 2D pose. This
part is described in Section 5.1. We will show that the re-
trieved poses are insufficient for estimating the 3D pose.
Instead, we estimate the pose by minimizing the projection
error under the constraint that the solution is close to the re-
trieved poses (Section 5.2). In addition, the retrieved poses
are efficiently retrieved by a graph

\[ \phi_{ij}(x_i, x_j) \]

where each vertex corresponds to 2D coordinates of a particular body joint \( i \) and \( j \). The binary potentials \( \phi_{ij}(x_i, x_j) \) define the deformation cost of joint \( i \) from its parent joint \( j \).

The unary potentials in (1) can be modeled by any dis-

\[ P(x|I) \propto \prod_{i \in \mathcal{J}} \phi_i(x_i|I) \prod_{(i,j) \in \mathcal{L}} \phi_{ij}(x_i, x_j), \]

where the unary potentials \( \phi_i(x_i|I) \) correspond to joint tem-

\[ \gamma_{ij}^c \exp \left( -
\frac{1}{2} \left( d_{ij} - \mu_{ij}^c \right)^T (\Sigma_{ij}^c)^{-1} \left( d_{ij} - \mu_{ij}^c \right) \right) \]

with mean \( \mu_{ij}^c \), covariance \( \Sigma_{ij}^c \) and \( d_{ij} = (x_i - x_j) \). The weights \( \gamma_{ij}^c \) are set according to the cluster frequency

\[ \frac{1}{c(i,j)^\alpha} \]

with a normalization constant \( \alpha = 0.1 \) [9].

5. 3D Pose Estimation

While the PSM for 2D pose estimation is trained on
the images with 2D pose annotations as shown in Fig. 1,
we now describe an approach that makes use of a second
dataset with 3D poses in order to predict the 3D pose from
an image. Since the two sources are independent, we first
have to establish relations between 2D poses and 3D poses.
This is achieved by using an estimated 2D pose as query for
3D pose retrieval (Section 5.1). The retrieved poses, how-
ever, contain many wrong poses due to errors in 2D pose
estimation, 2D-3D ambiguities and differences of the skele-
tons in the two training sources. It is therefore necessary
to fit the 3D poses to the 2D observations. This will be de-
scribed in Section 5.2.

5.1. 3D Pose Retrieval

In order to efficiently retrieve 3D poses for a 2D pose
query, we preprocess the motion capture data. We first nor-
malize the poses by discarding orientation and translation
information from the poses in our motion capture database.
We denote a 3D normalized pose with \( X \) and the 3D nor-
malized pose space with \( \Psi \). As in [33], we project the nor-
malized poses \( X \in \Psi \) to 2D using orthographic projection.
We use 144 virtual camera views with azimuth angles spanning 360 degrees and elevation angles in the range of
0 and 90 degree. Both angles are uniformly sampled with
step size of 15 degree. We further normalize the projected
2D poses by scaling them such that the y-coordinates of the
joints are within the range of \([-1, 1]\). The normalized 2D
pose space is denoted by \( \psi \) and does not depend on a spe-
cific camera model or coordinate system. This step is il-
lustrated in Fig. 1. After a 2D pose is estimated by the
approach described in Section 4, we first normalize it accord-
ing to \( \psi \), i.e., we translate and scale the pose such that the
y-coordinates of the joints are within the range of \([-1, 1]\),
then use it to retrieve 3D poses. The distance between two
normalized 2D poses is given by the average Euclidean dis-
tance of the joint positions. The K-nearest neighbours in
\( \psi \) are efficiently retrieved by a kd-tree [15]. The retrieved
normalized 3D poses are the corresponding poses in \( \Psi \). An
incorrect 2D pose estimation or even an imprecise estima-

Figure 2: Different joint sets. \( J_{up} \) is based on upper body
joints, \( J_{lw} \) lower body joints, \( J_{lt} \) left body joints, \( J_{rt} \) right
body joints and \( J_{all} \) is composed of all body joints. The
selected joints are indicated by the large green circles.
tion of a single joint position, however, can effect the accuracy of the 3D pose retrieval and consequently the 3D pose estimation. We therefore propose to use several 2D joint sets for pose retrieval where each joint set contains a different subset of all joints. The joint sets are shown in Fig. 2. While \( J_{all} \) contains all joints, the other sets \( J_{up}, J_{lw}, J_{lt} \) and \( J_{rt} \) contain only the joints of the upper body, lower body, left hand side and right hand side of the body, respectively. In this way we are able to compensate for 2D pose estimation errors, if at least one of our joint sets does not depend on the wrongly estimated 2D joints.

5.2. 3D Pose Estimation

In order to obtain the 3D pose \( \mathbf{X} \), we have to estimate the unknown projection \( \mathcal{M} \) from the normalized pose space \( \Psi \) to the image and infer which joint set \( J_s \) explains the image data best. To this end, we minimize the energy

\[
E(\mathbf{X}, \mathcal{M}, s) = \omega_p E_p(\mathbf{X}, \mathcal{M}, s) + \omega_r E_r(\mathbf{X}, s) + \omega_a E_a(\mathbf{X}, s)
\]

(3)

consisting of the three weighted terms \( E_p, E_r \), and \( E_a \).

The first term \( E_p(\mathbf{X}, \mathcal{M}, s) \) measures the projection error of the 3D pose \( \mathbf{X} \) and the projection \( \mathcal{M} \):

\[
E_p(\mathbf{X}, \mathcal{M}, s) = \left( \sum_{i \in J_s} \| \mathcal{M}(X_i) - x_i \| ^2 \right)^{\frac{1}{2}}, \tag{4}
\]

where \( x_i \) is the joint position of the predicted 2D pose and \( X_i \) is the 3D joint position of the unknown 3D pose. The parameter \( s \) defines the set of valid 2D joint estimates and the error is only computed for the joints of the corresponding joint set \( J_s \).

The second term enforces that the pose \( \mathbf{X} \) is close to the retrieved 3D poses \( \mathbf{X}_s^k \) for a joint set \( J_s \):

\[
E_r(\mathbf{X}, s) = \sum_k w_{k,s} \left( \sum_{i \in J_{all}} \| X_{s,i}^k - X_i \| ^2 \right)^{\frac{1}{2}}. \tag{5}
\]

In contrast to (4), the error is computed over all joints but the set of nearest neighbors depends on \( s \). In our experiments, we will show that an additional weighting of the nearest neighbors by \( w_{k,s} \) improves the 3D pose estimation accuracy.

Although the term \( E_r(\mathbf{X}, s) \) penalizes already deviations from the retrieved poses and therefore enforces implicitly anthropometric constraints, we found it useful to add an additional term that enforces anthropometric constraints on the limbs:

\[
E_a(\mathbf{X}, s) = \sum_k w_{k,s} \left( \sum_{(i,j) \in \mathcal{L}} (L_{s,i,j}^k - L_{i,j})^2 \right)^{\frac{1}{2}}, \tag{6}
\]

where \( L_{i,j} \) denotes the limb length between two joints.

Minimizing the energy \( E(\mathbf{X}, \mathcal{M}, s) \) (3) over the discrete variables \( s \) and the continuous variables \( \mathbf{X} \) and \( \mathcal{M} \) would be expensive. We therefore propose to obtain an approximate solution where we estimate the projection \( \mathcal{M} \) first. For the projection, we assume that the intrinsic parameters are given and only estimate the global orientation and translation. The projection \( \mathcal{M}_s \) is estimated for each joint set \( J_s \), with \( s \in \{up, lw, lt, rt, all\} \) by minimizing

\[
\mathcal{M}_s = \arg \min_{\mathcal{M}} \left\{ \sum_k E_p(\mathbf{X}_s^k, \mathcal{M}, s) \right\}
\]

(7)

using non-linear gradient optimization. Given the estimated projections \( \mathcal{M}_s \) for each joint set, we then optimize over the discrete variable \( s \):

\[
\hat{s} = \arg \min_{s \in \{up, lw, lt, rt, all\}} \left\{ \sum_k E(\mathbf{X}_s^k, \mathcal{M}_s, s) \right\}. \tag{8}
\]

As a result, we obtain \( \hat{s} \) and \( \mathcal{M} = \mathcal{M}_s \) and finally minimize

\[
\mathbf{X} = \arg \min_{\mathbf{X}} \left\{ E(\mathbf{X}, \mathcal{M}, \hat{s}) \right\}
\]

(9)
to obtain the 3D pose.

**Implementation details** Instead of obtaining \( \hat{s} \) by minimizing (8), \( \hat{s} \) can also be estimated by maximizing the posterior distribution for the 2D pose (1). To this end, we project all retrieved 3D poses to the image by

\[
x_{s,i}^k = \hat{\mathcal{M}}_s \left( \mathbf{X}_{s,i}^k \right).
\]

(10)
The binary potentials \( \phi_i,j(x_i, x_j | \mathbf{X}_s) \), which are mixture of Gaussians, are then computed from the projected full body poses for each set and \( \hat{s} \) is inferred by the maximum posterior probability:

\[
(\hat{x}, \hat{s}) = \arg \max_{\mathbf{X}, s} \left\{ \prod_{i \in \mathcal{J}} \phi_i(x_i | \mathbf{I}) \prod_{i,j \in \mathcal{L}} \phi_i,j(x_i, x_j | \mathbf{X}_s) \right\}. \tag{11}
\]

Finally, the refined 2D pose \( \hat{x} \) is used to compute the projection error \( E_p(\mathbf{X}, \mathcal{M}, \hat{s}) \) in (9).

In addition, we weight the nearest neighbors by

\[
w_{k,s} = \sum_{i \in \mathcal{J}} \phi_i(x_{s,i}^k | \mathbf{I}), \tag{12}
\]

to keep only the \( K_w \) poses with the highest weights, and normalize them by

\[
w_{k,s} = \frac{w_{k,s} - \min_{\mathbf{X}} (w_{k,s})}{\max_{\mathbf{X}} (w_{k,s}) - \min_{\mathbf{X}} (w_{k,s})}. \tag{13}
\]

The dimensionality of \( \mathbf{X} \) can be reduced by applying PCA to the weighted poses. We thoroughly evaluate the impact of the implementation details in Section 6.1.1.
5.3. Iterative Approach

The approach can be iterated by using the refined 2D pose \( \hat{\mathbf{s}} \) (11) as query for 3D pose retrieval (Section 5.1) as illustrated in Fig. 1. Having more than one iteration is not very expensive since many terms like the unaries need to be computed only once and the optimization of (7) can be initialized by the results of the previous iteration. The final pose estimation (9) also needs to be computed only once after the last iteration. In our experiments, we show that two iterations are sufficient.

6. Experiments

We evaluate the proposed approach on two publicly available datasets, namely HumanEva-I [23] and Human3.6M [13]. Both datasets provide accurate 3D poses for each image and camera parameters. For both datasets, we use a skeleton consisting of 14 joints, namely head, neck, ankles, knees, hips, wrists, elbows and shoulders. For evaluation, we use the 3D pose error as defined in [26]. The error measures the accuracy of the relative pose up to a rigid transformation. To this end, the estimated skeleton is aligned to the ground-truth skeleton by a rigid transformation and the average 3D Euclidean joint error after alignment is measured. In addition, we use the CMU motion capture dataset [8] as training source.

6.1. Evaluation on HumanEva-I Dataset

We follow the same protocol as described in [25, 14] and use the provided training data to train our approach while using the validation data as test set. As in [25, 14], we report our results on every 5th frame of the sequences walking (A1) and jogging (A2) for all three subjects (S1, S2, S3) and camera C1. For 2D pose estimation, we train regression forests and PSMs for each activity as described in [9]. The regression forests for each joint consists of 8 trees, each trained on 700 randomly selected training images from a particular activity. While we use \( c = 15 \) mixtures per part (2) for the initial pose estimation, we found that 5 mixtures are enough for pose refinement (Section 5.2) since the retrieved 2D nearest neighbors strongly reduce the variation compared to the entire training data. In our experiments, we consider two sources for the motion capture data, namely HumanEva-I and the CMU motion capture dataset. We first evaluate the parameters of our approach using the entire 49K 3D poses of the HumanEva training set as motion capture data. Although the training data for 2D pose estimation and the 3D pose data are from the same dataset, the sources are separated and it is unknown which 3D pose corresponds to which image.

Figure 3: (a) Using only joint set \( \mathcal{J}_s \). (b) Using all joint sets \( \mathcal{J}_s \) and estimating \( \hat{s} \) using (8). (c) All joint sets \( \mathcal{J}_s \) and estimating \( \hat{s} \) using (11).

Figure 4: Impact of number of nearest neighbors \( K \) and weighting of nearest neighbors \( K_w \). The results are reported for subject S3 with walking action (A1, C1) using the CMU dataset (a-b) and HumanEva (c-d) for 3D pose retrieval.

6.1.1 Parameters

Joint Sets \( \mathcal{J} \). For 3D pose retrieval (Section 5.1), we use several joint sets \( \mathcal{J}_s \) with \( s \in \{ \text{up, lw, lt, rt, all} \} \). For the evaluation, we use only one iteration and \( K = 256 \) without weighting. The results in Fig. 3 show the benefit of using several joint sets. Estimating \( \hat{s} \) using (11) instead of (8) also reduces the pose estimation error.

Nearest Neighbours. The impact of weighting the retrieved 3D poses and the number of nearest neighbours is evaluated in Fig. 4. The results show that the weighting reduces the pose estimation error independently of the used motion capture dataset. Without weighting more nearest neighbours are required. If not otherwise specified, we use \( K = 256 \) and \( K_w = 64 \) for the rest of the paper. If the average of the retrieved \( K \) or \( K_w \) poses is used instead of optimizing (9), the errors are 55.7mm and 48.9mm, respectively, as compared to 53.2mm and 47.5mm by optimizing (9). PCA can be used to reduce the dimension of \( X \). Fig. 5(a) evaluates the impact of the number of principal components. Good results are achieved for 10-26 components, but the exact number is not critical. In our experiments, we use 18.

Energy Terms. The impact of the weights \( \omega_r, \omega_p \) and \( \omega_a \) in (3) is reported in Fig. 5(b-d). Without the term \( E_r \), the
error is very high. This is expected since the projection error $E_p$ is evaluated on the joint set $J_p$. If $J_p$ does not contain all joints, the optimization is not sufficiently constrained without $E_r$. Since $E_r$ is already weighted by the image consistency of the retrieved poses, $E_p$ does not result in a large drop of the error, but refines the 3D pose. The additional anthropometric constraints $E_a$ slightly reduce the error in addition. In our experiments, we use $\omega_p = 0.55$, $\omega_r = 0.35$, and $\omega_a = 0.065$.

**Iterations.** We finally evaluate the benefit of having more than one iteration (Section 5.3). Fig. 6 compares the pose estimation error for one and two iterations. For completeness, the results for nearest neighbours without weighting are included. In both cases, a second iteration decreases the error on nearly all sequences. A third iteration does not reduce the error further.

**6.1.2 Comparison with State-of-the-art**

In our experiments, we consider two sources for the motion capture data, namely HumanEva-I and the CMU motion capture dataset.

**HumanEva-I Dataset.** We first use the entire 49K 3D poses of the training data as motion capture data and compare our approach with the state-of-the-art methods [14, 29, 25, 26, 6, 20]. Although the training data for 2D pose estimation and 3D pose data are from the same dataset, our approach considers them as two different sources and does not know the 3D pose for a training image. We report the 3D pose error for each sequence and the average error in Table 1. While there is no method that performs best for all sequences, our approach outperforms all other methods in terms of average 3D pose error. The approaches [14, 6] achieve a similar error, but they rely on stronger assumptions. In [14] the ground-truth information is used to compute a 3D bounding volume and the inference requires around three minutes per image since the approach

![Figure 5: (a) Impact of the number of principal components. The error is reported for subject S3 with action jogging (A2, C1) using the CMU dataset for 3D pose retrieval. (b-d) Impact of the weights $\omega_r$, $\omega_p$, and $\omega_a$ in (3).](image)

![Figure 6: Impact of number of iterations and weighting of nearest neighbors.](image)
Table 1: Comparison with other state-of-the-art approaches on the HumanEva-I dataset. The average 3D pose error (mm) and standard deviation are reported for all three subjects (S1, S2, S3) and camera C1. * denotes a different evaluation protocol. (a) Results of the proposed approach with one or two iterations and motion capture data from the HumanEva-I dataset. (b) Results with motion capture data from the CMU dataset.

Table 2: Impact of MoCap data. (a) MoCap from HumanEva dataset. (b) MoCap from HumanEva dataset without walking sequences. (c) MoCap from HumanEva dataset but skeleton is retargeted to CMU skeleton. (d) MoCap from CMU dataset. The average 3D pose error (mm) is reported for the HumanEva-I dataset with one iteration.

Table 3: 2D pose estimation error (pixels) after refinement.

Table 4: Comparison on the Human3.6M dataset. (a) 2D pose estimated as in Section 4 (b) 2D pose from ground-truth. (c) MoCap dataset includes 3D pose of subject S11.

6.2. Evaluation on Human3.6M Dataset

The protocol originally proposed for the Human3.6M dataset [13] uses the annotated bounding boxes and the training data only from the action class of the test data. Since this protocol simplifies the task due to the small pose variations for a single action class and the known scale, a more realistic protocol has been proposed in [14] where the scale is unknown and the training data comprises all action classes. We follow the protocol [14] and use every 64th frame of the subject S11 for testing. Since the Human3.6M dataset comprises a very large number of training samples, we increased the number of regression trees for 2D pose estimation to 30 and the number of mixtures of parts to c = 40, where each tree is trained on 10K randomly selected training images. We use the same 3D pose error for evaluation and perform the experiments with 3D pose data from Human3.6M and the CMU motion capture dataset. In the first case, we use six subjects (S1, S5, S6, S7, S8 and S9) from Human3.6M and eliminate very similar 3D poses. We consider two poses as similar when the average Euclidean
Figure 7: Four examples from HumanEva-I. From left to right: estimated 2D pose \( x \) (Section 4); retrieved 3D poses from all joint sets (Section 5.1); retrieved 3D poses from inferred joint set \( J_\hat{s} \) (Section 5.2); retrieved 3D poses weighted by \( w_{k,\hat{s}} \), \( \hat{s} \) (13); refined 2D pose \( \hat{x} \) (11); estimated 3D pose \( \hat{X} \) (9) shown from two different views.

Figure 8: Comparison on the Human3.6M dataset.

7. Conclusion

In this paper, we have presented a novel dual-source approach for 3D pose estimation from a single RGB image. One source is a MoCap dataset with 3D poses and the other source are images with annotated 2D poses. In our experiments, we demonstrate that our approach achieves state-of-the-art results when the training data are from the same dataset, although our approach makes less assumptions on training and test data. Our dual-source approach also allows to use two independent sources. This makes the approach very practical since annotating images with accurate 3D poses is often infeasible while 2D pose annotations of images and motion capture data can be collected separately without much effort.
Acknowledgements. Hashim Yasin gratefully acknowledges the Higher Education Commission of Pakistan for providing the financial support. The authors would also like to acknowledge the financial support from the DFG Emmy Noether program (GA 1927/1-1) and DFG research grant (KR 4309/2-1).

References

[8] CMU. Carnegie mellon university graphics lab: Motion capture database, 2014. mocap.cs.cmu.edu. 1, 5, 6
[23] L. Sigal, A. O. Balan, and M. J. Black. Humanev: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. IJCV, 2010. 1, 5