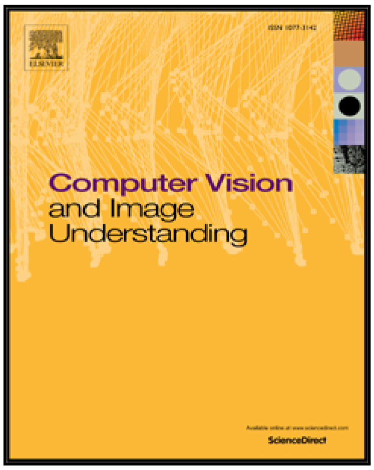
A local-global coupled-layer puppet model for robust online human pose tracking

Ma, Miao; Marturi, Naresh; Li, Yibin; Stolkin, Rustam; Leonardis, Ales

DOI:
10.1016/j.cviu.2016.08.010

Accepted Manuscript



A Local-Global Coupled-Layer Puppet Model for Robust Online Human Pose Tracking

Miao Ma, Naresh Marturi, Yibin Li, Rustam Stolkin, Ales Leonardis

PII: S1077-3142(16)30118-7
DOI: 10.1016/j.cviu.2016.08.010
Reference: YCVIU 2473

To appear in: *Computer Vision and Image Understanding*

Received date: 31 August 2015
Revised date: 28 May 2016
Accepted date: 22 August 2016

Please cite this article as: Miao Ma, Naresh Marturi, Yibin Li, Rustam Stolkin, Ales Leonardis, A Local-Global Coupled-Layer Puppet Model for Robust Online Human Pose Tracking, *Computer Vision and Image Understanding* (2016), doi: 10.1016/j.cviu.2016.08.010

Highlights

- We propose a new method for online tracking of articulated human body poses.
- Our method offers online sequential tracking from one frame to the next.
- Many other methods mutually optimize poses offline over all frames of a sequence.
- We propose a novel cross-coupled global-local model of articulated human body pose.
- We propose an adaptive penalty function for optimizing the pose estimates.

ACCEPTED MANUSCRIPT

A Local-Global Coupled-Layer Puppet Model for Robust Online Human Pose Tracking

Miao Ma[a,b,*], Naresh Marturi[b,c], Yibin Li[a], Rustam Stolkin[b], Ales Leonardis[b]

[a]*Shandong University, Jinan, Shandong, 250061, P.R.China*
[b]*University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK*
[c]*KUKA robotics UK Ltd., Wednesbury Great Western Street, WS10 7LL, UK*

Abstract

This paper addresses the problem of online tracking of articulated human body poses in dynamic environments. Many previous approaches perform poorly in realistic applications: often future frames or entire sequences are used anti-causally to mutually refine the poses in each individual frame, making online tracking impossible; tracking often relies on strong assumptions about *e.g.* clothing styles, body-part colours and constraints on body-part motion ranges, limiting such algorithms to a particular dataset; the use of holistic feature models limits the ability of optimisation-based matching to distinguish between pose errors of different body parts. We overcome these problems by proposing a coupled-layer framework, which uses the previous notions of deformable structure (DS) puppet models. The underlying idea is to decompose the global pose candidate in any particular frame into several local parts to obtain a refined pose. We introduce an adaptive penalty with our model to improve the searching scope for a local part pose, and also to overcome the problem of using fixed constraints. Since the pose is computed using only current and previous frames, our method is suitable for online sequential tracking. We have carried out empirical experiments using three different public benchmark datasets, comparing two variants of our algorithm against four recent state-of-the-art (SOA) methods from the literature. The results suggest comparatively strong performance

*Corresponding author
Email address: mamiaosdu@hotmail.com (Miao Ma)

of our method, regardless of weaker constraints and fewer assumptions about the scene, and despite the fact that our algorithm is performing online sequential tracking, whereas the comparison methods perform mutual optimisation backwards and forwards over all frames of the entire video sequence.

Keywords: human pose tracking, human tracking, video tracking, pose estimation, coupled-layer model.

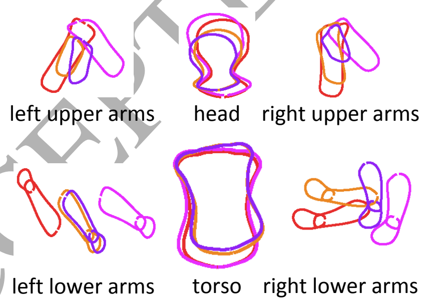
---

1. Introduction

Human pose estimation and tracking are increasingly popular research areas in computer vision, and have been studied for well over 30 years in the literature, *e.g.* [1]. There is growing interest in such algorithms for a variety of applications including activity recognition [2], video understanding [3], gesture analysis [4], human-robot interaction [5], and others. Significant advances were made in recent years, however even state-of-the-art (SOA) methods often rely on strong assumptions and constraints in representing human bodies, such as visual appearance [4], scale [6], lighting conditions, occlusions, and the ranges of motion of limbs and limb-parts. In this work, our goal is to sequentially track human body poses in monocular video frames obtained under variable conditions, where people move freely and interact with each other. Typical examples include videos of TV series or movies, where human appearance is unconstrained (*e.g.* variable background, any colour and type of clothing, no fixed scale, etc.). Many recent efforts have been devoted to track and estimate human poses from monocular video frames. Even though most of them perform well on certain body parts such as torsos and heads, their performance for arms is still not convincing. Within this context, we are most closely interested in tracking upper body poses, which include head, torso and arms, and in particular, improving the pose accuracy of lower arms. Nevertheless, our approach is not constrained for human upper body and can be easily adapted to the entire body. Our method is initialised from a single frame, and does not require any prior knowledge of the human clothing style, background scene or other conditions.

A variety of methods have been proposed in recent years to track and estimate the poses of articulated human bodies. However, many methods make use of the entire image sequence to mutually refine the poses in each individual frame, *e.g.* [7, 8], rendering them only suitable for offline applications. In contrast, our method relies only on the previous frame information at any point in time, with computation only in the temporal direction, enabling online tracking applications. Since this reduction in available temporal information affects the overall performance, our method makes use of additional information from the spatial domain. For estimating articulated human pose, the overall information associated with the target makes the state space too large to compute. In this case, we exploit a local-global coupled-layer method, which uses the entire human body as a global layer and uses decomposed parts as a local layer (see Fig. 1). This type of methodology not only reduces the computational space and cost, but also improves the overall accuracy.

(a) Global candidates.



(b) Local candidates

(c) Refined human pose

Figure 1: Proposed coupled-layer model. (a) Different global pose candidates; (b)Local parts obtained by decomposing the global pose candidates. (c) Recomposed global pose.

In this paper, we present an on-line coupled-layer method using discrete-structure puppets [9] for estimating the upper human body pose information. Recently published human pose estimation methods predominantly use an evaluation function to evaluate a candidate pose for the entire human body [10, 11]. However, such methods can become prone to local convergence problems. For example, if one candidate pose suggests a correct left arm position, and an erroneous right arm position, and an alternative candidate pose is vice versa, then both candidates may generate similar evaluation scores. In this paper, we address this problem by decomposing the entire body into smaller parts and by estimating the pose separately for each of them. Nevertheless, if enough constraints are not provided, this decomposition method will also be unreliable, *e.g.* left and right arms may erroneously swap places and converge on each other's true image locations. To resolve this issue we introduce an adaptive penalty policy (Sec. 4.3.3) with our coupled-layer method to improve the scope of local parts pose searching. It also assists in tackling variable body scales and tuning any propagated erroneous poses.

The remainder of this paper is organized as follows. The methods that are closely related to our work are presented in section 2. The proposed coupled-layer model is presented in section 3, where we detail the model and explain the relationship between its local and global layers. Section 4 explains the tracking and estimation procedure, using the coupled-layer model. Section 5 presents experiments conducted using three different public benchmark datasets, where we compare the performance of our method against four other SOA pose estimation techniques. In this section, we also investigate the robustness of our method to various different levels of initialization error. Section 6 concludes the paper and the proposed method.

2. Related Work

Numerous human pose estimation techniques, developed for a variety of applications, are available in the literature. In this section, we discuss the work most closely related to our proposed method.

The well-known *pictorial structures* (PS) model, proposed by Fischler and Elschlager [12] in 1973, is still drawing significant attention from researchers for its efficient tree-based inference algorithm [11, 10, 13, 14, 15]. A key limitation of PS, and some extended models, is that the parts are treated as rigid templates and are represented as rectangular (or polygonal) regions. Later methods, such as *contour people* [16] and *deformable structures* (DS) model [9], that are derived from 3D human models, can better capture the 2D shape as non-rigid, deformable parts. However, due to the holistic nature of these models, several problems can arise e.g. in the case of rapid part motions or occlusions.

Several methods from the literature use some kind of hierarchical methodology or coarse-to-fine scheme for inference. For example, Wu and Huang [17] used a two-layer model for hand motion tracking, where the palm motion is represented in the global model and the fingers motion in the local model. Kuo *et al.* [18] used a two-layer model which searches for the coarse location of the human body regions over the image sequence in one layer, and then estimates and refines detailed human body part poses over the image sequence in another layer. Lee and Nevatia [19] proposed a three-layer model. An alternative strategy is to model each part separately [20, 21, 22] and impose different constrains on different parts [23]. However, these methods estimate and evaluate the entire body together. Related works such as [24] and [7] focus on individual body parts *i.e.* to treat a single lower arm or an entire limb as an independent part to explore a set of poses. However, in such work, the entire video sequence is typically used to mutually refine the poses over all images, making them unsuitable for online tracking. In contrast, in this paper we propose a local-global coupled strategy, in which poses are tracked in an online fashion from one frame to the next using a holistic body model for the global layer (Fig. 1(a)), while refining poses within each frame using individual body part models as the local layer (Fig. 1(b)).

In some pose estimation methods, optical flow information is exploited as a cue, either for body part detection or for frame-to-frame pose propagation.

Zuffi *et al.* [8] use both forwards and backwards optical flow to propagate pose. The major drawback of this approach is that it cannot be used for online tracking. Additionally, the accuracy of such methods is limited unless applied to a particular dataset, because the joint angle space is pre-constrained to match the limited range of poses appearing in a particular video sequence. This makes the method difficult to adapt to more varied datasets, or real world applications with changing or uncertain scenes. Fragkiadaki *et al.* [25] have used kinematically constrained optical flow for segmenting body parts and for propagating segmentations over time. Cherian *et al.* [7] made use of the optical flow between current and future frames to create loops for passing messages. The messages passed within these loops then help to constrain the location of each node. Similar to these methods, we also use optical flow in this work for both pose estimation and propagation. However, we additionally exploit an adaptive penalty policy which automatically constrains the searching space instead of fixing it in advance (particular to a given dataset) or using future information (offline mutual refining of poses over all frames of a sequence).

Sometimes occlusions and self-occlusions occur in unconstrained environments, and such situations are difficult to handle. In 3D tracking, Cho *et al.* [26] solved this problem by modeling self-occlusion states between two body parts utilizing the 3D pose information of each body part (modeled as 3D cylinders). However in 2D conditions, it is much harder to obtain depth information for helping to detect occlusion states. Chen and Yuille [27] indirectly solved this problem using an image dependent pairwise relational term for adjacent body parts. In contrast, our work proposes an adaptive penalty policy, which makes it possible to predict the possible location of a body part under occlusion, and also enables the re-detection and tracking of the body part when it re-appears following a period of occlusion.

A common schema for human pose estimation is, firstly, generating a number of pose candidates, then constructing a reliable cost function as well as making a non-maximum suppression (NMS) method to find the most likely human pose. Sigal and Black [28] used a hierarchical method which need enough plausible pose part candidates for belief propagation. Park and Ramanan [15] proposed a method to generate a diverse set of N-best candidate poses with small overlaps for a still image, depending on a large number of pose hypotheses generated using the method of [29]. Later, Cherian [7] decomposed the N-best candidates generated by [15] and recomposed them using information from all frames of the entire video sequence to find refined poses. Burgos *et al.* [30] define a loss function for the large number of predicted pose candidates, with respect to time and space for all frames of the entire sequence, and use the scores of the loss function to decide a final pose for each frame. In the work of Zuffi *et al.* [8], the NMS method is also used to generate a good initial estimate among numerous pose candidates for each still image. In this schema, the NMS method relies on information derived from all frames of the entire video sequence, which limits these methods only for offline applications, and also requires that the set of candidate poses is large enough to contain "good" poses for each frame. In contrast, our method does not rely on large numbers of extra pose candidates generated for each image. We only use a small number of whole body candidates in our global layer, and after decomposing global candidates into local candidates, our method is able to relocate keypoints to get additional local candidates and refine them online using only information from only the current frame and one previous frame.

3. Proposed Coupled-Layer Model using DS Puppets

The DS (deformable structures) puppet model is a 2D articulated human body model recently introduced by Zuffi *et al.* [9], and applied to human pose tracking and estimation in [8]. The human's shape is expressed as a factored probability over parts [9]. The DS puppets model is learned from training contours derived from SCAPE [31] (Shape Completion and Animation of People), which is a parametric 3D model of articulated human shape. Our method is also based on the DS puppet, however, we decompose it into multiple layers (local and global) for estimating the final pose. Hence, we call our model a *coupled-*

layer DS puppet model. In our case, we use the model that has been trained using SCAPE while the testing is performed using SOA datasets explained in detail in Sec.5.1. The performed experiments point towards the generality and independence of the model.

Our coupled-layer model is inspired by the *local-global tracker* (LGT) [32], where a single target object, defined by a simple bounding box, is tracked by combining feature models (*e.g.* colour histograms, motions and shapes) for the overall object (global layer) and several small patches (the local layer). Each layer is used to help constrain (and thereby robustify) updates for the other layer. Our proposed articulated pose estimation method adopts a similar philosophy. As shown in Fig. 1, for a certain frame t, our method operates in three successive stages: procure global layer puppet, handle individual local layer parts and estimate refined global pose. The local layer contains groups of every upper body part and each group is comprised of several pose candidates. The process to initialise and select best pose candidates is detailed in Sec. 4. The global layer has nine keypoints to generate the entire human upper body, and in the similar fashion to local layer it has its own global pose candidates. In each frame, the entire global upper body poses are decomposed into local body parts, from which the local layer refines each part separately and filters out bad candidates. The refined local parts are re-combined into global layer candidates for further processing within the global layer.

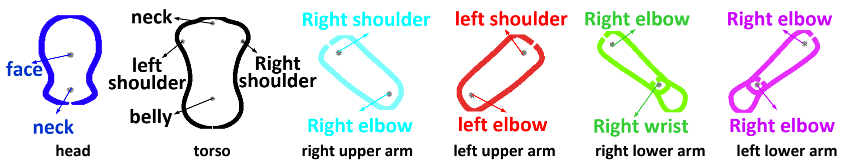
In Sec. 3.1 and Sec. 3.2 we describe the composition of local and global layers, respectively and in Sec. 3.3 we provide an overview of the local-global coupled-layer puppet model.

3.1. Local Layer

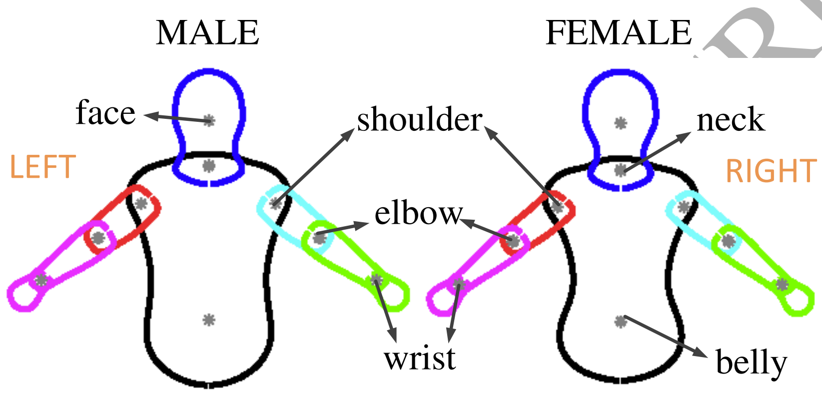
The local layer $\mathcal{L}$ in the t^{th} frame is composed of 6 parts as follows:

$$\mathcal{L}_t = \{H_t, T_t, UA_t^r, UA_t^l, LA_t^r, LA_t^l\}, \tag{1}$$

where, H and T denote head and torso, UA^r and UA^l stand for right and left upper arms, LA^r and LA^l represent right and left lower arms, respectively (see



(a) Local parts with their keypoints $\mathbf{k}_i^{(j)}$, described in Eq.(2). For torso, $\mathbf{k}_i^{(j)}$ includes four keypoints while for other parts, $\mathbf{k}_i^{(j)}$ includes two keypoints.



(b) Global poses with their keypoints

Figure 2: Illustration of the keypoints in local and global layers. (a) Keypoints of each part present in the local layer of a female puppet. (b) Keypoint locations of the global upper body male and female puppets. It can be seen that every part has two keypoints, some of them also belong to other parts (*e.g.* neck, left/right elbows).

Fig. 1(b)). These six parts are the main body parts of the upper human body and contain vital human body pose information. For simplicity and sequential calculation, hereafter we maintain the same order for parts given in Eq.(1) throughout this work. Each individual part P_i ($i = 1 \cdots 6$ with 1 for head, 2 for torso and so on as in Eq.(1)) is specified by three elements:

$$P_i = \{\mathbf{k}_i^{(j)}, s_i^{(j)}, model_i\}_{j=1:N_i}, \tag{2}$$

where, N_i is the number of candidates of part i, $\mathbf{k}_i^{(j)}$ is the keypoints location of the j^{th} candidate in part i, see Fig. 2(a). For torso, $\mathbf{k}_i^{(j)}$ includes four keypoints while for other parts, $\mathbf{k}_i^{(j)}$ includes two keypoints. $s_i^{(j)}$ is the scale of this local layer candidate, which is inherited from the *scale* of global layer (scale computation is demonstrated in Sec. 3.2 and illustrated in Fig. 3) and $model_i$ is the model of part i used to calculate the part candidate closed contour $\mathcal{C}_i^{(j)}$. This model has been obtained through the principal component analysis (PCA)-based method proposed by [9]. It contains a vector $\mathbf{m}_i$ representing the mean contour and keypoints of part i, and a matrix $\mathbf{B}_i$ containing the eigenvectors of the training data corresponding to the dominant eigenvalues, for each gender separately. For the reason that females and males require different models, the principal components are trained separately for both genders.

The relationship among $\mathbf{k}_i^{(j)}$, $s_i^{(j)}$, $\mathcal{C}_i^{(j)}$ and $model_i$ is shown in Eq.(3):

$$\begin{bmatrix} \mathcal{C}_i^{(j)} \\ \mathbf{k}_i^{(j)}, s_i^{(j)} \end{bmatrix} = \mathbf{B}_i \mathbf{z}_i^{(j)} + \mathbf{m}_i, \tag{3}$$

where, $\mathbf{z}_i$ is a vector of linear shape coefficient. Given $\mathbf{k}_i^{(j)}$ and $s_i^{(j)}$, we can calculate $\mathbf{z}_i^{(j)}$ according to Eq.(3). With fixed $\mathbf{z}_i^{(j)}$, the contour $\mathcal{C}_i^{(j)}$ of the j^{th} local candidate can be calculated.

3.2. Global Layer

The global layer $\mathcal{G}$ is able to estimate the shape and scale of the entire upper body and to connect the selected candidates of each part from the local layer in order to estimate the overall human body pose. Each global candidate in layer $\mathcal{G}$ has 9 keypoints $\mathcal{K}$ (shown in Fig. 2(b)) as follows:

$$\mathcal{K} = \{belly, face, neck, rsh, re, rw, lsh, le, lw\}, \tag{4}$$

where rsh/lsh mean right/left shoulders, re/le mean right/left elbows, and rw/lw mean right /left wrists. The global contour $\mathcal{GC}$ of the q^{th} candidate in t^{th} frame is given by:

$$\mathcal{GC}_t^{(q)} = \bigcup_{i=\{i|i\subset\mathcal{L}_t\}} \mathcal{C}_i^{(q)}. \tag{5}$$

Each scale $s_i^{(q)}$ used to calculate $\boldsymbol{\mathcal{C}_i^{(q)}}$ is of the same value with *scale*, which is described later in this section. Similar to the layer $\mathcal{L}$, different models for males and females are used in this layer as shown in Fig. 2(b). Each global layer pose candidate has a probability $p(\boldsymbol{\mathcal{GC}}_t^{(q)}|\pi_{DS})$ according to the DS puppet defined in [8] (π_{DS} refers to DS model parameters), which represents the probability of a global model instance.

Here, we exploit a method to estimate the global model scale using defined keypoints $\boldsymbol{\mathcal{K}}$. We find that the most invariant relative distance d_c of the keypoints is:

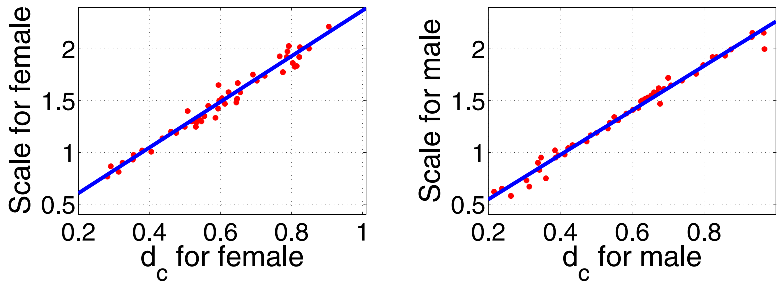
$$d_c = d_{(neck,face)} + d_{(neck,lsh)} + d_{(neck,rsh)}. \tag{6}$$

Eq. (6) gives the sum of the Euclidean distances between neck and head, and neck and left/right shoulders. In this context, we use "Transfer Learning" [33] to obtain a relationship between d_c and scale. This has been accomplished using 50 static images for each gender that are obtained from online image databases containing arbitrary human poses (with varying scale). For each image, we define a set of keypoints to calculate the d_c value (see Fig. 3(a)) and a corresponding scale value. Now, the obtained d_c and scale values will guide us in estimating a linear relationship as shown in Fig. 3(b). Since males and females require different body models, separate male and female sequences are used for training. Consequently, a global body puppet contour has been obtained in the first frame from Eq.(5) using nine keypoints, as shown in Fig. 3(c).

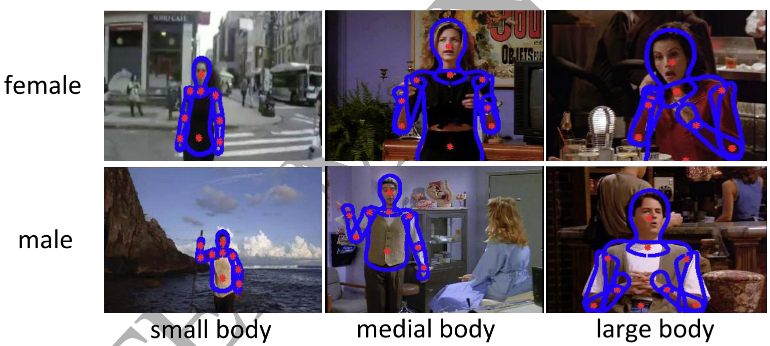
3.3. Overview of the Proposed Coupled-layer Model

A schematic overview of the proposed coupled-layer model is depicted in Fig. 4. In order to estimate the human body pose in frame $t+1$, initially we propagate several best entire pose candidates estimated in frame t to frame $t+1$ according to optical flow (illustrated in Fig. 4 step1) which will be described in Sec. 4.2. Then we use a flexible mixtures of parts (FMP) method [10], which is a human pose estimation method for monocular still images, to generate several extra entire human pose candidates for frame $t+1$ (Fig. 4 step2). This step is performed to provide more options when locating torsos. At this point, we have propagated candidates and initialised candidates (from FMP) in the global layer as shown in Fig. 1(a), and in the next step (Fig. 4 step3) we decompose them into

(a) Sample external images with pre-defined keypoints.



(b) Estimated linear relationship between scale and d_c.



(c) Suitable scale obtained from the relationship shown in (b).

Figure 3: (a) Sample images used for scale computation, first two show female body keypoints and the next two show male body keypoints. (b) and (c) Illustration of scale and global puppet estimation. (b) Relationship between d_c and scale, dots represent training samples. (c) Obtained initial frame global body puppets with different scales, dots represent keypoints.

local layer candidates (see Fig. 1(b)) for further processing. To refine these local layer candidates, we use a method described in Sec. 4.4 to generate additional relocated local part candidates when necessary (Fig. 4 step4). After this step, a cost function defined in Sec. 4.3 is used to select best local part candidates, which

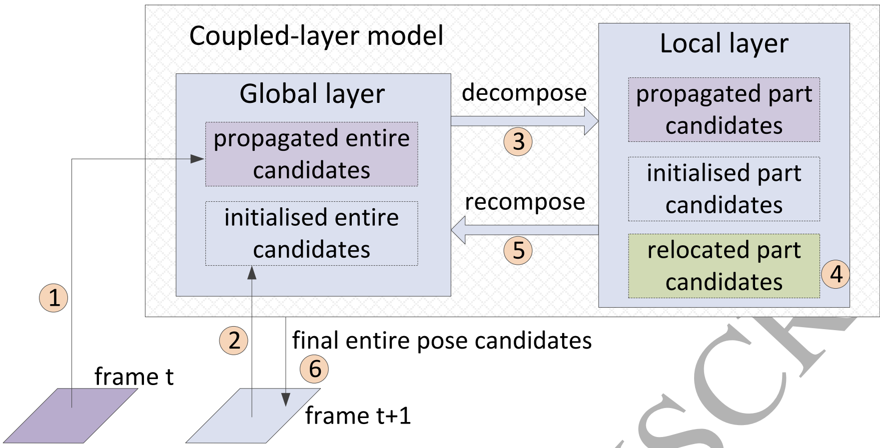


Figure 4: A schematic overview of *coupled-layer DS puppet model* for the frame $t+1$. There are several steps: 1) propagate several best global human pose candidates from frame t to frame $t+1$; 2) generate several entire pose candidates using FMP method for the frame $t+1$; 3) decompose all the global layer candidates into local part candidates; 4) generate some relocated local part candidates when necessary; 5) recompose selected local parts into global candidates; 6) get final best entire human pose candidates for frame $t+1$.

are later recomposed into global entire human pose candidates (Fig. 4 step5). Then we evaluate the recombined global candidates (Sec. 4.1), and choose the best candidates to propagate to frame $t+2$ for future pose estimation (Fig. 4 step6). The best candidate is selected as the overall result of frame $t+1$ (see Fig. 1(c)).

4. Inference

4.1. Body Pose Initialization

Our method does not use any posterior information (unlike [8] which uses forwards and backwards temporal propagation), and the available knowledge about each part is limited. To resolve this problem, some researchers have assumed prior knowledge such as the colour of the tracked person's clothes [8] or a predetermined start pose, and others, e.g. [34], assume a manual initialization at the first frame (similar to conventions of the mainstream target tracking literature). In this work we follow the latter approach by defining the puppet manually in the first frame of the video sequence. This is accomplished by selecting nine keypoints of a human body (*e.g.* belly button, neck, face, etc. that are defined in Eq.(4)), and then Eq.(5) is used to obtain the initial global pose (Fig. 3(c)).

People often wear coloured clothes (either with long or short sleeves) and this colour information can be used for recognition and tracking. In our method, we extract colour histograms $h_c(i)$ for each local part i from the first frame, handling self-occlusion from lower arms to upper arms, and then to torso and head. The RGB image frames are transformed into the CIE L*a*b* colour space, and the pixels which have very small Lightness values ($L < 0.3$) are ignored. The two colour dimensions (a and b) and 20×20 bins are used to calculate the colour histograms $h_c(i)$. Later, this information is used for matching in the local layer (as presented in Sec. 4.3.1).

4.2. Global Layer Pose Tracking

Due to the possibility of erroneous hand-initialised poses (or, in future applications, erroneous automatic detections) in the first frame, we perturb the initialised pose to obtain several global pose candidates. As discussed in Sec. 3.3, after processing each frame, we get several global pose candidates for propagation. We calculate the score of each global layer candidate, based on which the best candidates for propagation are selected. In our method, the best 8 candidates are selected for propagating to the next frame. The score for any q^{th} global candidate in the t^{th} frame is computed as follows:

$$score_t^{(q)} = \psi_t^{(q)} + \phi_t^{(q)} = \lambda_\psi p(I_t|\boldsymbol{\mathcal{GC}}_t^{(q)}) + \lambda_\phi p(\boldsymbol{\mathcal{GC}}_t^{(q)}|\pi_{DS}), \tag{7}$$

where the coefficients $\lambda_\psi >> \lambda_\phi$ for the reason that the magnitude of $\phi_t^{(q)}$ is larger than $\psi_t^{(q)}$. The first term $\psi_t^{(q)} = p(I_t|\boldsymbol{\mathcal{GC}}_t^{(q)})$ contains the image likelihood (*i.e.* colour and contour likelihood) for the entire puppet, I_t is the t^{th} frame of video sequence, and $\boldsymbol{\mathcal{GC}}_t^{(q)}$ is the q^{th} whole puppet candidate contour for the current frame. The second term in Eq. (7), $\phi_t^{(q)}$ (defined in [8]) represents the probability of a DS model instance. We assume that the set of best poses in frame t are approximately correct, and we then track the whole body poses from frame t to $t+1$ using the optical flow of each part region of frame t. The optical flow images are computed using the method proposed by Liu [35]. Next, we calculate an affine matrix $\boldsymbol{A}_i^{(q)}$ (an affine motion model proposed by [8]) for each individual part i within the candidate q, which is used to estimate displacements of keypoints $\boldsymbol{\mathcal{K}}$. Because some keypoints may lie at the intersection region of two different parts, the final displacement for such keypoints is approximated by the mean of that found for each part. The keypoint displacements are calculated as

$$vp_k^{(q)} = \frac{1}{N_k} \sum_{i=\{i|k\subset part\ i\}} \tilde{vp}_{k,i}^{(q)}, \quad in\ which\ \tilde{vp}_{k,i}^{(q)} = \boldsymbol{A}_i^{(q)} \tilde{\boldsymbol{k}}_i^{(q)}, \tag{8}$$

where $\tilde{\boldsymbol{k}}_i^{(q)}$ is the regularized keypoints[1] location in part i of the q^{th} entire upper body candidate. $\tilde{vp}_{k,i}^{(q)}$ is the displacements of the keypoints k in part i of the q^{th} global candidate according to the optical flow. $N_k = 1$ if the keypoint k belongs to only one part (*e.g.* head and belly button); otherwise $N_k = 2$ (*e.g.* shoulder and elbow), as illustrated in Fig. 2.

In addition to the propagated candidates from the previous frame, in order to improve accuracy in estimating the torso and head locations, we use the FMP method [10] to add a few additional candidates to the propagated candidates, as shown in Fig. 4 step2.

4.3. Local Layer Pose Estimation

After generating a set of global upper body pose candidates, we need to decompose them into local layer parts, in order to refine each part separately. Each local layer candidate acquires a scale $s_i^{(j)}$ from the *scale* of the related global layer candidate. We refine each local layer part in the same sequence as defined in Eq. (1).

[1] $\tilde{\boldsymbol{k}}_i^{(q)}$ are used along with the affine matrix $\boldsymbol{A}_i^{(q)}$ to fit an affine motion model to the optical flow matrix within each body part.

A cost function $p(I_{t+1}|\boldsymbol{C}_i^{(j)})$ is used to evaluate every candidate of each part in the local layer separately:

$$p(I_{t+1}|\boldsymbol{C}_i^{(j)}) = \lambda_1 p_{ct}(I_{t+1}|\boldsymbol{C}_i^{(j)}) + \lambda_2 p_{cl}(I_{t+1}|\boldsymbol{C}_i^{(j)}) + \lambda_3 p_p(I_{t+1}, I_t|\boldsymbol{C}_i^{(j)}) \\ + \lambda_4 p_f(I_{t+1}, I_t|\boldsymbol{C}_i^{(j)}) + \lambda_5 p_h(I_{t+1}, I_t|\boldsymbol{C}_i^{(j)}). \tag{9}$$

The cost function considers five factors. In the first two terms we consider image likelihood, where we use contour $p_{ct}(I_{t+1}|\boldsymbol{C}_i^{(j)})$ and colour $p_{cl}(I_{t+1}|\boldsymbol{C}_i^{(j)})$. The next term is our adaptive penalty $p_p(I_{t+1}, I_t|\boldsymbol{C}_i^{(j)})$, automatically adapts constraint terms while estimating limb locations (in contrast to [8] which limits joint angles to match the motion range of a particular dataset, or [25] which imposes a-priori kinematic constraints). The remaining two parts relate to motion likelihood, which are motion cue $p_f(I_{t+1}, I_t|\boldsymbol{C}_i^{(j)})$ and hand motion offset $p_h(I_{t+1}, I_t|\boldsymbol{C}_i^{(j)})$. Because of the magnitude of the five terms, the selection of corresponding parameters should be $\lambda_3 < 0 < \lambda_4 < \lambda_5 \leq \lambda_2 < \lambda_1$. Fig. 5 illustrates various scores of different part candidates given by the cost function. It is evident that the highest score provides the best candidate.

Figure 5: Illustration of the discriminative power of the cost function.

4.3.1. Image likelihood

Firstly, we describe how to calculate contour likelihood $p_{ct}(I_{t+1}|\boldsymbol{C}_i^{(j)})$. The scale of human bodies varies greatly within different video sequences, as shown in Fig. 3(c). To make the contour-based likelihood more robust, similar to [8], we use a three-level pyramid to apply a histogram of oriented gradients (HOG) descriptor: at the contour, inside the contour, and outside the contour, in order to obtain a feature vector $h_i(I_{t+1}|\mathcal{C}_i^{(j)})$. Next, a support vector machine (SVM) classifier is applied to this feature vector to compute $p_{ct}(I_{t+1}|\mathcal{C}_i^{(j)})$.

$$p_{ct}(I_{t+1}|\mathcal{C}_i^{(j)}) = \frac{1}{1+\exp\left(a_i\, svm\left(h_i(I_{t+1}|\mathcal{C}_i^{(j)})\right)+b_i\right)}, \tag{10}$$

where the function $svm(\cdot)$ means the output of the SVM, a_i and b_i are scalar parameters [36]. The SVM is trained on a collected dataset (217 images) with annotations as shown in [9].

Next, the colour histograms $h_c(i)$ previously computed for individual parts (Sec. 4.1) are now used to generate a colour probability map $M_c(i)$ (considering self-occlusion) for each part, as illustrated using an instance of a lower arm part in Fig. 6. We handle the self-occlusion by masking other parts in an order from lower arms to upper arms, and then to torso and head. We use the first propagated puppet of frame *t+1* to handle the self-occlusion, in case that the masked parts would not influence the evaluation of part *i*. By checking the value of each pixel within $\mathcal{C}_i^{(j)}$ in $M_c(i)$, we calculate the mean value of these pixels as colour-based likelihood $p_{cl}(I_{t+1}|\mathcal{C}_i^{(j)})$.

4.3.2. Motion likelihood

We compute a motion image F_{t+1}, *i.e.* optical flow from frame t to frame *t+1*, as shown in Fig. 6. When handling the motion image for each part, we consider the self-occlusions among parts in a similar way with the method used in Sec.4.3.1, but we also mask the other parts regions of the puppet from frame t, because the F_{t+1} is calculated using both frame t and *t+1*.

The motion image F_{t+1} is masked for each part candidate, and a flow region $region_i^{(j)}$ for part i in the j^{th} candidate can be computed. Then, the motion-based likelihood $p_f(I_{t+1}, I_t|\mathcal{C}_i^{(j)})$ is calculated as the mean value of pixels within this region.

$$p_f(I_{t+1}, I_t|\mathcal{C}_i^{(j)}) = \tfrac{1}{N}\textstyle\sum_{(x,y)\subset region_i^{(j)}} F_{t+1}(x,y), \tag{11}$$

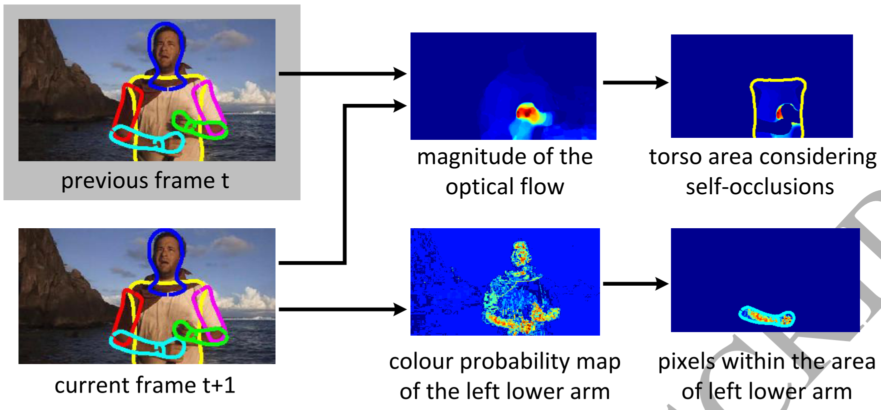


Figure 6: Illustration of the colour probability map and the optical flow magnitude. The images of frame *t* and *t+1* are on the **left**. The **upper middle** image shows the magnitude of the optical flow from frame *t* to *t+1*, and the **upper right** image shows the magnitude of optical flow for torso area considering self-occlusions. The **lower middle** image reveals the colour probability for the colour of left lower arm area, and the **lower right** image shows the colour probability map pixels within the area of left lower arm.

where, N is the total number of pixels within $region_i^{(j)}$, I_{t+1} and I_t are images corresponding to the frames $t+1$ and t, respectively, and $\boldsymbol{\mathcal{C}}_i^{(j)}$ is the index of the j^{th} local candidate of part i defined in Eq.(3).

Hands (the distal regions of left/right lower arm parts) tend to be more flexible and move faster than other parts, and so should not have the same penalty as other parts. We therefore add the motion-based item only for lower arms ($i \subset \{LA_t^r, LA_t^l\}$) to offset some of the penalty. We generate a hand motion map $H_{t+1} = f_h(I_{t+1}, I_t)$ for each frame by using a hand filter [6] over optical flow gradient magnitude. Masking H_{t+1} to get pixels within the hand region $Mask_i^{(j)}$, and the mean value of these pixels is used to build $p_h(I_{t+1}, I_t|\boldsymbol{\mathcal{C}}_i^{(j)})$:

$$p_h(I_{t+1}, I_t|\boldsymbol{\mathcal{C}}_i^{(j)}) = \frac{1}{N}\sum_{(x,y)\subset Mask_i^{(j)}} H_{t+1}(x,y). \tag{12}$$

4.3.3. Adaptive penalty

In general, estimating the pose for each part separately may lead to low efficiency and unexpected failures. To overcome this problem, we introduce an adaptive penalty function. We start by computing the displacement value $vp_k^{(q)}$ of each keypoint (denoted by k) in the q^{th} global candidate during propagation (see Eq.(8)), and record the maximum and minimum values as boundaries. Then we choose a movement vc_k (between the maximum and minimum) of keypoint k as:

$$vc_k = \min_{1 \leqslant q \leqslant N_q}(vp_k^{(q)}) + \lambda_v(\max_{1 \leqslant q \leqslant N_q}(vp_k^{(q)}) - \min_{1 \leqslant q \leqslant N_q}(vp_k^{(q)})), \tag{13}$$

where, λ_v is a fixed coefficient, and $\lambda_v \in (0,1)$. We also set keypoint movement $v_{k,i}^{(j)}$ to be the displacement of k in the j^{th} local candidate of part i from I_t to I_{t+1}, and the difference between vc_k and $v_{k,i}^{(j)}$ is denoted by $ve_k^{(j)}$. We define the coarse penalty term as follows:

$$\tilde{p_p}(I_{t+1}, I_t|\boldsymbol{\mathcal{C}}_i^{(j)}) = \sum_{k=\{k|k \subset part\ i\}}(||ve_k^{(j)}||_2), \tag{14}$$

where I_{t+1} and I_t refers to images in frames $t+1$ and t, respectively, and $\boldsymbol{\mathcal{C}}_i^{(j)}$ means the index of the j^{th} local candidate of part i defined in Eq.3.

Human lower arms sometimes move fast, and human body parts frequently self-occlude or may be occluded by other objects. Consider a situation when a local part location in frame t is erroneous due to an occlusion, and the occluded body part re-appears in the next frame. In this case the penalty term in Eq.(14) may cause problems when the local part needs to correct its pose by rapidly jumping from the wrong (old) location to the new location of the reappeared part. Our global layer overcomes this problem.

In the global layer, the score, $score_1^{(1)}$ (calculated using Eq.(7)) is recorded when manually initialising the puppet in the first frame, and $score_{t+1}^{(1)}$ is calculated after propagating from frame t to frame $t+1$. Additionally, we set a threshold for penalty as $D_p = \frac{d_c}{2}$, where d_c is defined in Eq.(6). Then revisiting the local layer, we define our adaptive penalty as follows:

$$p_p(I_{t+1}, I_t|\mathcal{C}_i^{(j)}) = \begin{cases} (\frac{1}{\omega \cdot D_p}) \cdot \tilde{p_p}(I_{t+1}, I_t|\mathcal{C}_i^{(j)}), & if\ \tilde{p_p}(I_{t+1}, I_t|\mathcal{C}_i^{(j)}) \leqslant D_p \\ \frac{1}{\omega} & , otherwise \end{cases}, \tag{15}$$

where, $\omega = \begin{cases} \frac{score_1^1 - score_{t+1}^1}{|score_1^1|}, & \omega \geqslant \delta \\ \delta & , otherwise \end{cases}$, $D_p = \frac{d_c}{2}$, δ is a small positive value which is set to be 0.1, and $\tilde{p_p}(I_{t+1}, I_t|\mathcal{C}_i^{(j)})$ is the coarse penalty term defined in Eq.(14).

4.4. From Decomposition to Recomposition

After refining local parts, the next step in our method is to recombine all local parts to form a global refined pose. Previously, Yang and Ramanan [10] used a tree model-based method for calculating over all parts iteratively to get the best configuration for the position and type of each root. Later, they generate multiple detections in each image. By tracking the *argmax* indices, they find the location and type of each part in each maximal configuration. Our selection for the best part candidates is different from such methods and is explained below.

As mentioned earlier, we follow the same order mentioned in Eq. (1) for pose computation and now for re-composition we follow the reverse order *i.e.* to calculate from lower arms to torso and head. The hand colour and motion maps can be used to sample the possible wrist locations. However, if the sampled wrist is too far from the elbow (further than the predefined lower arm length threshold), the elbow needs be relocated to make sure the lengths of both upper and lower arm are within the required range. In this process, we search for a new elbow location along the detected lower arm direction, while ensuring that the lower arm length meets the length constraint. This process also results in new upper arm candidates.

From all the sampled, propagated and initialised results, the cost function defined in Eq.(9) is used to obtain a best set of lower arm candidates $\boldsymbol{N}_{la}$. Next, relocated elbows from the previous step result in new upper arm candidates. From all relocated, propagated and initialised upper arms, the best set of upper arm candidates $\boldsymbol{N}_{ua}$ are also selected using Eq.(9).

Once we have both upper and lower arm candidates, the next step is to find the complete right and left arms by connecting $\boldsymbol{N}_{ua}$ and $\boldsymbol{N}_{la}$. Each upper and

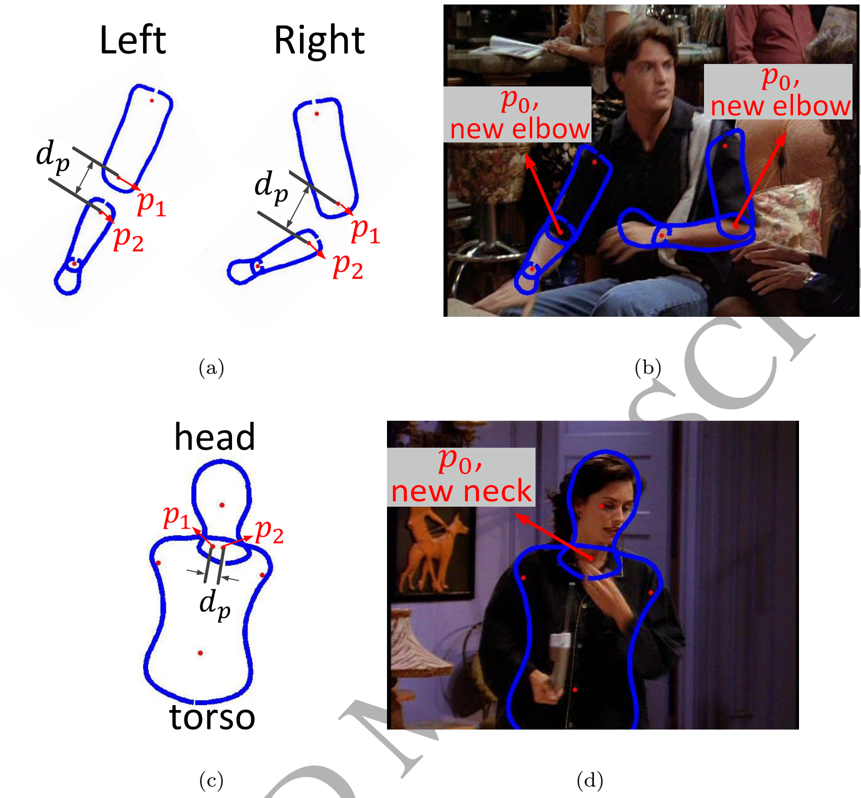


Figure 7: (a) Left and right arm candidates with upper (p_1) and lower (p_2) elbow points. (b) Connected new elbow point p_0. (c) Head and torso candidates with neck (p_1 and p_2) points. (d) Connected new neck point p_0.

lower arm candidate contains an elbow point (*i.e.* p_1 and p_2 in Fig. 7(a)). The process is performed in two steps. Initially, the upper and lower arm candidates are classified into pairs with the smallest Euclidean distance d_p between p_1 and p_2 to represent various complete arms (Fig. 7(a) shows one pair for left side and one for right side as examples). In the process of pairing, each half arm (lower or upper) can be used more than once to ensure every half arm could find its nearest other half. Secondly, a final elbow location p_0 is obtained using Eq.(16). The threshold τ in Eq.(16) is used to judge whether or not the two parts are too far away from each other.

$$p_0 = \begin{cases} \frac{p_1+p_2}{2} , & \text{if } d_p < \tau \\ p_1 + \frac{1}{10} \cdot d_p , & \text{otherwise} , \end{cases} \tag{16}$$

where, $\tau = \tau_0 \cdot scale$, and τ_0 is a threshold of pixel distance which is set in Table.2 of Sec.5.2.1. p_0 is the new connecting joint point, as illustrated in Fig. 7(b) and (d).

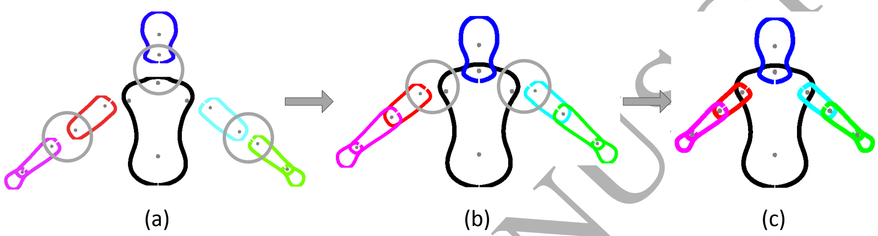


Figure 8: The procedure of connecting local part candidates to obtain a refined global pose.

Head and torso pair sets, and torso and left/right upper arm pair sets are selected in the same way. The procedure for connecting local part candidates is shown in Fig. 8. In each case the two parts are connected by calculating new left/right shoulders and new necks, respectively using Eq.(16). When calculating new necks, the points p_1, p_2 and p_0 are defined as in Fig. 7(c) and (d); when calculating new shoulders, p_1 refers to the shoulder point on the torso while p_2 refers to the point on the upper arm, and p_0 refers to the calculated new shoulder point for the connected torso - upper arm pair. Note that, before calculating new shoulders, heads are already connected with torsoes and left/right lower arms are already connected with upper arms, as shown in Fig. 8(b). Once new shoulders are calculated, the entire bodies are obtained, as shown in Fig 8(c).

Next, we return to the global level $\mathcal{G}_{t+1}$ and use Eq.(7) to obtain several best puppet bodies for propagation to the next frame $t+2$ (as illustrated in Fig. 4 step1 and discussed in Sec. 4.2). The best global pose candidate is selected as the final pose for the current frame $t+1$.

Algorithm 1 Local-Global Coupled-Layer Upper Body Pose Tracker.

Choose $\boldsymbol{\mathcal{K}}$.
Generate global human pose $\boldsymbol{\mathcal{GC}}$.
Perturb $\boldsymbol{\mathcal{GC}}$ to get N_p global candidates.
for $t = 2, 3, 4...$ **do** ▷ t means frame index.
    Propagate N_p $\boldsymbol{\mathcal{GC}}$s to frame t, and generate N_i $\boldsymbol{\mathcal{GC}}$s using FMP.
    Decompose each $\boldsymbol{\mathcal{GC}}$ into P $\boldsymbol{\mathcal{C}}_i^{(j)}$s. ▷ P is the number of parts in $\mathcal{L}_t$.
    In LA_t^r and LA_t^l, search for new $\boldsymbol{rw}$s and $\boldsymbol{lw}$s, and adjust $\boldsymbol{re}$s and $\boldsymbol{le}$s, which lead to new LA_t^r, LA_t^l, UA_t^r and UA_t^l.
    for $i = 1$ to P **do**
        Select best $\boldsymbol{\mathcal{C}}_i^{(j)}$s using Eq.(9).
    end for
    Make UA_t^r and LA_t^r, UA_t^l and LA_t^l, H_t and T_t into pairs.
    Connect each pair using p_0.
    Connect arms to torsos by calculating p_0 of rsh and lsh, to get $\boldsymbol{\mathcal{GC}}$s.
    Select best N_p $\boldsymbol{\mathcal{GC}}$s using Eq.(7).
end for

4.5. Implementation Analysis

We implement the above presented method in Matlab running on a Windows 7 machine with 3.4 GHz Intel i5 CPU. The key steps are summarised in Algorithm.1. Since the method is online, its complexity depends on the number of candidates N and number of parts P to process in the current image. In its current form of implementation, the corresponding asymptotic time complexity is computed to be of $\mathcal{O}(PN)$, where $N = N_p + N_i$. Currently, it takes 4 seconds to process an image and estimate the pose.

5. Experiments

5.1. Datasets Description and Evaluation Methodology

Three different public benchmark datasets have been used for evaluation experiments. The *VideoPose2.0-training* dataset (we didn't use this dataset for

Figure 9: Sample frames of our experimental datasets. (a) Frames from *VideoPose2.0-training* dataset, (b) frames from *VideoPose2.0-testing* dataset, and (c) frames from *Pose in the Wild* dataset.

training - only for testing) and *VideoPose2.0-testing* dataset, which contain 26 clips and 18 clips respectively (each clip has about 30 frames), are obtained from two popular TV series "Friends" and "Lost" [6]. Our experiments use all sequences of the *VideoPose2.0-training* dataset, referred to as *VideoPose-1*, see Fig. 9(a), and *VideoPose2.0-testing* dataset, referred to as *VideoPose-2*, see Fig. 9(b). Additionally, we use *Pose in the Wild* dataset [7], a challenging dataset which has 30 sequences extracted from the Hollywood movies "Forrest Gump", "The Terminal", and "Cast Away". Each sequence has about 30 frames with widely changing or deforming body poses. We refer to this dataset as *WildPose*, see Fig. 9(c).

Some well known work, such as [7], evaluate and report their results by recording the percentage of keypoints that lie within a threshold number of pixels $error_o$ from the ground truth. However human images in different video sequences have different scales, which makes it unfair and unmeaningful to use a constant number of pixels to evaluate the estimation error, as shown in Fig. 10(a). Therefore, similar to the other SOA methods e.g. [8], we introduce a normalized set of threshold number of pixels (*pixels error*) $error_r$ as follows:

$$error_r = error_o \times scale, \tag{17}$$

(a) a set of threshold numbers of pixels

(b) a set of normalized threshold numbers of pixels, calculated by Eq.(17)

Figure 10: Un-normalized and normalized threshold number of pixels. Six circles stand for six thresholds, from inside to outside which has 15, 20, 25, 30, 35, 40 pixels radius, respectively. (a) Un-normalized thresholds are too small for the left (large scale) figure but too large for the right (small scale) figure. (b) Normalized thresholds are much more meaningful for frames of different scales.

where, *scale* is illustrated by Fig. 3 in Sec. 3.2. This yields more meaningful evaluation results, as demonstrated in Fig. 10(b). For each frame in every sequence, the *scale* in Eq.(17) is stored with the ground truth for repeating experiments, and each method reported in Fig. 11 is evaluated in the same way using Eq.(17). Fig. 11 plots the elbow and wrist accuracy of each method, averaged over all frames of all sequences of the respective dataset. The reported elbow/wrist accuracy is the mean accuracy value of the left and right elbow/wrist. The horizontal axis in Fig. 11 is the pixels error $error_o$ used in Eq.(17).

5.2. Discussion of Human Pose Estimation Results

In this subsection, we first compare two variants of our method (*i.e.* with and without the adaptive penalty term) against four SOA methods, as described in Sec. 5.2.1. Then in Sec. 5.2.2, we evaluate the robustness of our proposed method.

5.2.1. Comparison experiments

Here we present an experimental evaluation of our coupled-layer method where we compare two different versions of our method against the SOA methods of Zuffi *et al.* [8], Sapp *et al.* [6], Cherian *et al.* [7], as well as Park and Ramanan [15]. The adaptive motion penalty is a critical part of our proposed method. To demonstrate its significance, two different runs are performed with each dataset: one with the penalty and the other without.

To perform these comparisons, we used the source code provided by Zuffi *et al.* [8] and Cherian *et al.* [7] for their methods to carry out the experiments on all datasets. When using the same datasets as used in the comparison papers, we use parameters as reported by the authors; while for different datasets, we used modified parameters that are chosen using the same methodology proposed by the corresponding work. For the methods of Sapp *et al.* [6] and Park and Ramanan [15], due to the lack of access to their source code, we compare our method against their previously published results with the same public datasets.

Note that these comparisons are non-trivial. The problem of "detecting" a human (and its pose) in a single image, is a separate and distinct computer vision problem to that of sequentially tracking a human from one frame to the next. However, many published studies combine both these computer vision problems/methods in a single work, so that the two techniques (detection and tracking) can become confounding factors for evaluating the performance of either. The compared methods are not "online" in that they apply a moderately weak (noisy) pose detector to all frames over an entire video sequence, and then mutually optimise the poses, backwards and forwards, across all frames to satisfy smoothness and mutual compatibility constraints. In contrast, our method is "online" in the sense that it only makes use of information from the preceding frame, to estimate the pose in the current frame. Since our method relies on no prior knowledge except the estimated pose at the previous frame, it would not be fair or meaningful to initialise using a weak or noisy pose detector at the first frame, and we therefore hand-initialise our tracker in the first frame. To ensure a persuasive comparison, we use the same hand-initialised poses in the first frame of each sequence when we evaluate the methods of Zuffi *et al.* [8] and Cherian *et al.* [7] (the results are shown in Fig. 11). We suggest that the compared methods represent the best of the available SOA methods for human pose estimation in video sequences, and it is therefore useful and sensible to show comparison of these "offline" methods against our own "online" method in this paper. We believe that our use of identical hand-initialised poses for the first frame of all compared methods, makes for a fair comparison. Additionally, we note that: i) we have observed that the use (or not) of hand-initialised ground-truth for the first frame of the compared techniques makes very little difference to their performance (unsurprising, since the compared methods rely on separate detections in all frames); ii) in the next section we investigate the sensitivity of our proposed method to varying levels of noise in the initial pose estimate, and find it to be relatively robust against such perturbations.

The first row in Fig. 11 shows the experimental results of all methods tested on the *VideoPose2.0-training* dataset. Results of Fig. 11(a) suggest that the proposed coupled-layer method with adaptive penalty provides significantly better elbow localization accuracy than [7] and [8], by 16% and 18% respectively at 15 *pixels error*, and this superiority is maintained until 40 *pixels error*. Fig. 11(b) shows that the wrist accuracy of our method is around 20% better than [7] and [8] over all *pixels error* thresholds. One possible explanation for the lower performance of Zuffi *et al.* [8] on this dataset, is that they assume the lower arm to be of skin colour, *e.g.* people wear semi-sleeve shirts. However only 54% clips in this dataset comply with this condition. Cherian *et al.* [7] have high requirements of the candidates, but the method they used to obtain pose candidates requires that some frames in the video sequences provide easy to detect poses. In the *VideoPose2.0-training* dataset, people sometimes wear loose clothes with long sleeves and self occlusion often occurs, which limits the accuracy of pose candidates and could be a possible factor to explain the lower accuracy of [7].

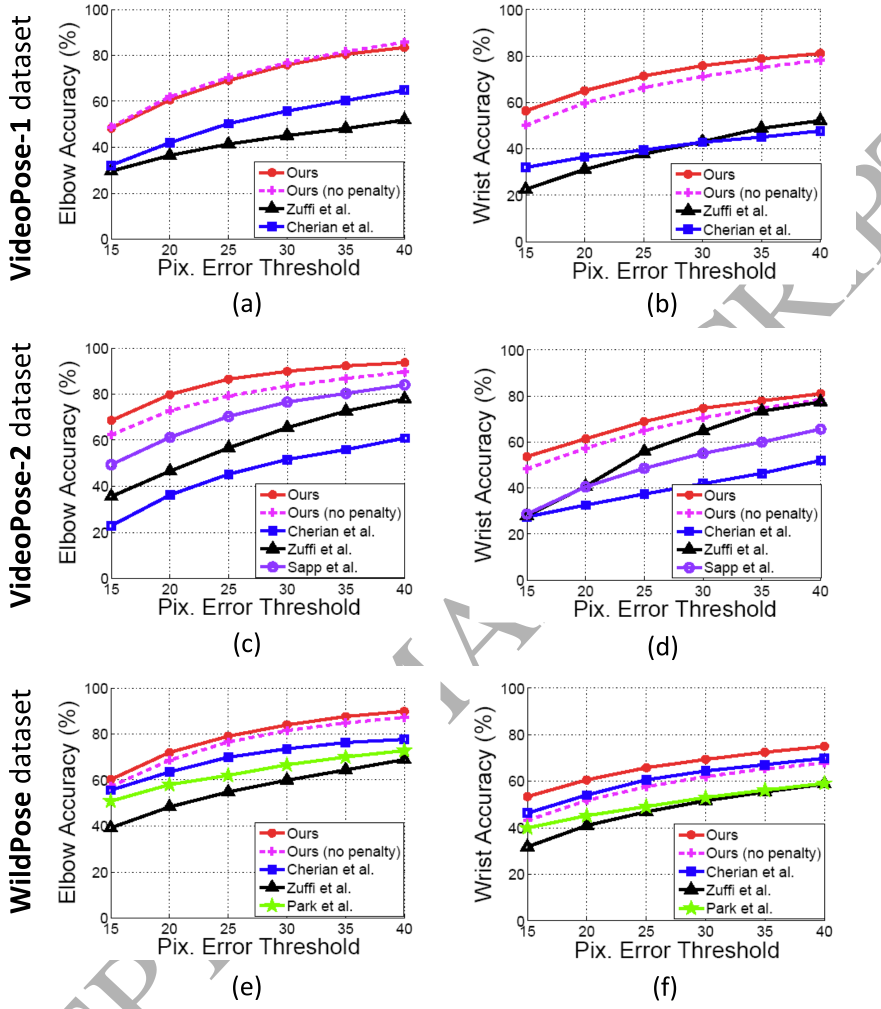


Figure 11: Performance comparison of the proposed method, with and without adaptive penalty, versus other SOA methods.

Fig 11(c) shows that our method clearly outperforms the SOA work of [8, 6] and [7] on elbow accuracy tested on the *VideoPose2.0-testing* dataset. From Fig. 11(d) we can see that performance accuracy is better than [8, 6, 7] by more than 20% at 15 *pixels error*. Then as *pixels error* is increased, Zuffi *et al.* method [8] improves comparatively. This is mainly due to the fact that all the poses are iteratively propagated and refined (forwards and backwards) within the entire video sequence, even if this results in losing the correct pose in many frames. However, this is the major advantage of our method, where a misjudged wrist pose in one frame can be corrected directly in the next frame using the proposed adaptive penalty.

The *WildPose* dataset is very different from the *VideoPose2.0* dataset. It contains more difficult outdoor scenes, with cluttered backgrounds, larger and faster movements of the tracked person, and rapid camera motion. The human poses are closer to those of real world scenarios. Our proposed method, with adaptive penalty term, significantly outperforms the comparison methods [7, 15] and [8] at all *pixels error* tolerances, on both elbow and wrist metrics, as presented in Fig. 11(e) and (f). This suggests that such offline learning-based methods, requiring the entire video sequence to be mutually refined over all poses in all frames, perform poorly in these challenging conditions compared to the more highly constrained conditions of the *VideoPose2.0* data. The performance of [8] is especially poor, likely due to their use of stronger assumptions and constraints (*e.g.* upper arm and torso should be of similar colour).

Table 1: Comparison of shoulder accuracy data

Datasets and Methods		Shoulder accuracy at x *pixels error* (%)					
		x=15	x=20	x=25	x=30	x=35	x=40
VideoPose-1	ours	**65.9**	**79.6**	**87.6**	**91.5**	**93.2**	**94.0**
	[8]	22.8	35.8	48.5	61.6	68.3	72.1
	[7]	63.8	68.6	71.2	72.6	73.8	74.9
VideoPose-2	ours	**69.2**	**82.2**	**88.8**	**91.4**	**93.4**	**95.0**
	[8]	30.4	58.9	79.1	90.1	95.8	96.5
	[7]	63.7	72.1	75.5	77.5	78.3	79.1
WildPose	ours	**56.0**	**71.0**	**81.5**	**87.7**	**91.1**	**93.4**
	[8]	34.9	49.9	63.7	74.2	79.7	84.0
	[7]	66.3	76.1	79.9	81.8	83.5	84.7

Torso locations are most likely to represent overall human position, which is, in turn, the foundation for estimating articulated human pose. Here we also compare our shoulder accuracy (see Table.1) with the SOA methods of [8] and [7]. Table.1 reveals that our method significantly outperforms other SOA methods in terms of accuracy of torsos.

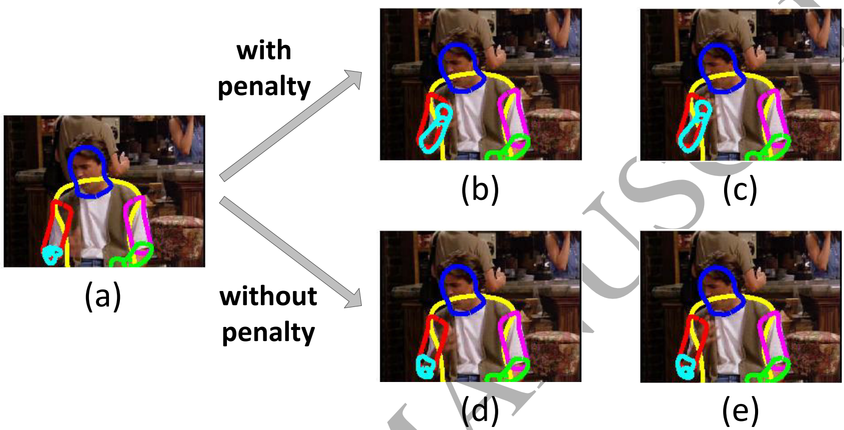


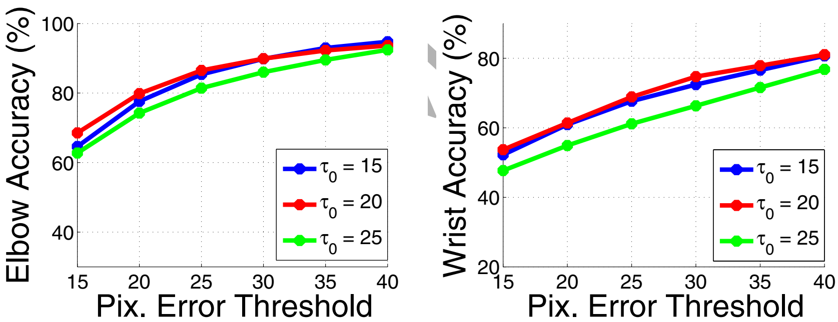
Figure 12: Performance analysis of using adaptive penalty. From the same frame with pose (a), poses (b) and (c) are achieved with penalty term, while poses (d) and (e) are achieved without penalty term. It can be clearly seen that the estimation performance is better using the penalty term.

Additionally, note that the advantage of using the adaptive penalty term with our coupled-layer method is clearly noticeable in all experiments of Fig. 11. Fig. 12 shows some examples to illustrate how the adaptive penalty term is able to improve pose tracking accuracy.

The parameter values used to test the method and their corresponding selection criteria are summarized in Table.2. Among these parameters, only τ_0 in Eq.(16) has been hand selected (constant) for the sake of implementation convenience. However, we vary its value and test our method on the *VideoPose2.0-testing* dataset in order to find the sensitivity of method to τ_0. Fig. 13 illustrates the resulting tracking accuracy for various τ_0 values. These results demonstrate that our proposed method is not sensitive to varying the value of τ_0. The values

Table 2: List of the parameters used in the experiments and corresponding selection criteria.

Equation	Coefficients	Selection
global candidates score Eq.(7)	λ_ψ=1, λ_ϕ=0.03	$\lambda_\psi >> \lambda_\phi$
local part candidates score Eq.(9)	λ_1=4, λ_2=1, λ_3=-0.6, λ_4=0.5, $\lambda 5$=1	$\lambda_4 < \lambda_5 \leq \lambda_2 < \lambda_1$, $\{\lambda_1, \lambda_2, \lambda_4, \lambda_5\} \in R_{>0}$, $\lambda_3 \in R_{<0}$
global layer keypoint movement Eq.(13)	λ_v=2/3	$0 < \lambda_v < 1$
relocate new keypoint Eq.(16)	τ_0=20.	$\tau_0 < 25$, not sensitive, see Fig. 13



(a) (b)

Figure 13: Proposed method is not sensitive to varying values of τ_0. (a) elbow accuracy when varying τ_0; (b) wrist accuracy while varying τ_0.

of the parameters reported in Table.2 are fixed for all our experiments *i.e.*, for all the sequences of all three datasets.

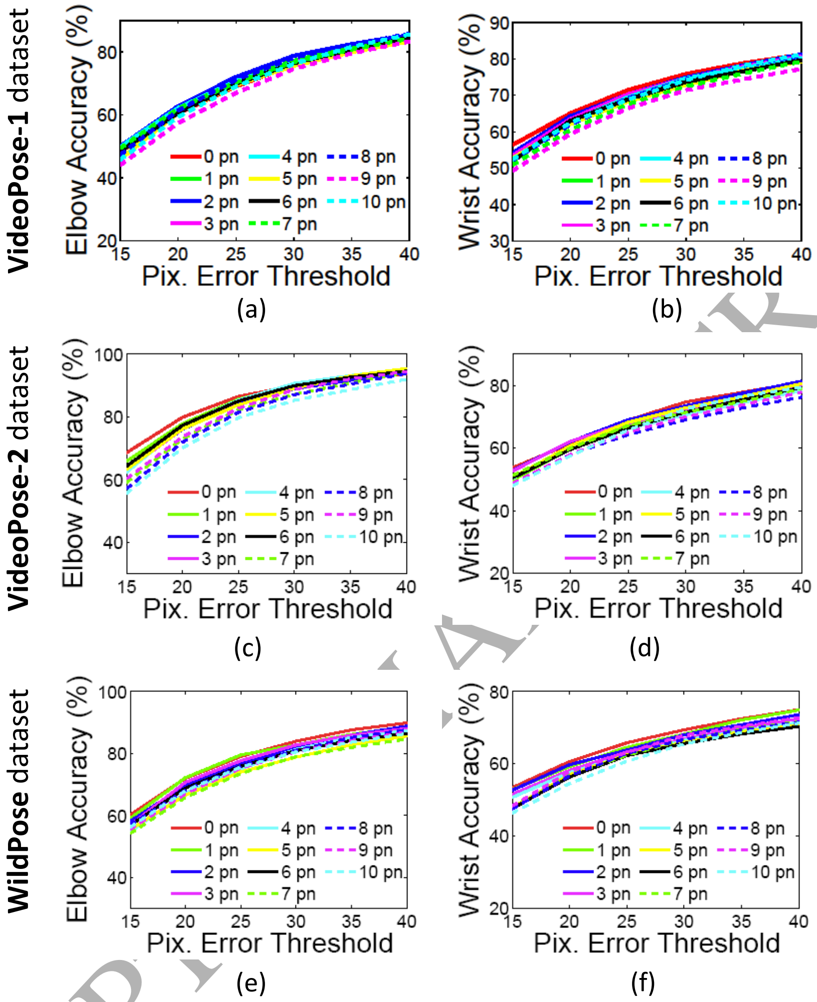


Figure 14: Results of using Gaussian noise to perturb the hand-initialised pose for the first frame of every video sequence. The amplitude of Gaussian noise ranges from 1 to 10 pixels. The unit 'pn' in legend means *pixel noise*, which refers to the amplitude of Gaussian noise.

5.2.2. Robustness experiments

To investigate the robustness of our method to varying levels of noise in the initial pose estimates at the first frame, we add noise to perturb these manually initialised poses, and use these perturbed poses to initialise our method. We perturb the ground-truth (manually initialised) poses by applying Gaussian

Table 3: Robustness for Initialization

Datasets and Joint Points		Accuracy with n pixels amplitude Gaussian noise (%)				
		n=0	n=4	n=7	n=10	*average*
VideoPose-1	*sh*	**91.5**	91.0	90.6	89.2	90.5
	el	**75.9**	76.5	76.8	76.0	76.7
	wr	**75.8**	74.6	72.4	74.2	74.0
VideoPose-2	*sh*	**91.4**	92.3	92.0	92.3	92.2
	el	**89.9**	90.7	87.2	85.3	88.7
	wr	**74.7**	71.0	71.4	72.5	72.2
WildPose	*sh*	**87.7**	86.8	85.6	85.3	85.6
	el	**83.0**	81.3	79.0	80.5	81.0
	wr	**69.4**	68.0	66.7	65.6	67.1

In this table, *sh* means *shoulders*, *el* means *elbows*, and *wr* means *wrists*. *average* means the average accuracy value among n ranges from 1 to 10.

noise, with amplitudes varying from 1 pixel to 10 pixels. We perturb the first frame pose for *VideoPose2.0-testing* dataset, *VideoPose2.0-testing* dataset and *Pose in the Wild* dataset separately. Fig. 14 shows the accuracy results for both elbow and wrist of each dataset, and Table.3 shows instance accuracy of shoulders, elbows and wrist for different amplitudes of Gaussian noise at 30 *pixels error*. The average accuracy of joint points among adding Gaussian noise from 1 to 10 pixels is also shown in Table.3. It can be seen that the added noise in the initial frame does not noticeably affect performance. This suggests that our method is robust to noisy initial pose estimates in the first frame. This phenomenon further supports the validity of the previous section which compares the performance of our tracker against SOA methods which rely on separate detections at each frame (see previous discussion of this).

Furthermore, we also demonstrate our method using the automatic initialization technique shown in [10]. We perform this test using the *VideoPose2.0-*

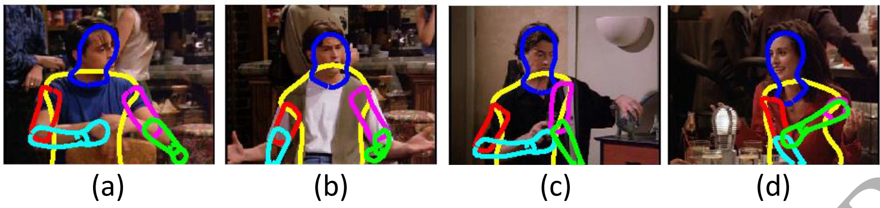
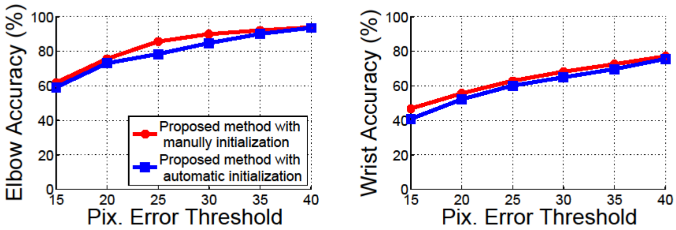
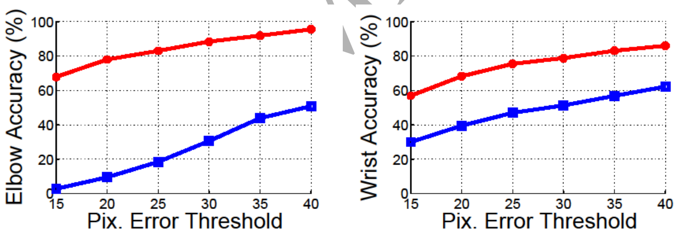


Figure 15: Samples of automatic initialization in the first frame. (a) and (b) show samples of acceptable auto-initialization; (c) and (d) show wrong auto-initialization, which cannot give correct information to the system.



(a) Implemented with acceptable automatic initializations



(b) Implemented with bad automatic initializations

Figure 16: Results of our proposed method with automatic initialization in the first frame. (a) shows result obtained by implementing with acceptable auto-initialization; (b) shows result obtained by implementing with wrong auto-initialization.

testing dataset, where the human body pose in the first frame has been automatically initialised. The dataset contains 18 clips, out of which the automatic initialization was acceptably successful for 12 clips and performed poorly for the rest, as shown in Fig. 15. Obtained accuracies in both cases are shown in Fig. 16. As expected, the results show that the proposed pose tracker works reasonably well in the case of effective initialization. In contrast, in cases where the automatic initialization failed, then successive tracking has difficulty in recovering from the very large initial errors. This is due to the fact that the proposed method does not rely on any prior knowledge, while the automatic initialization fails to give correct target information.

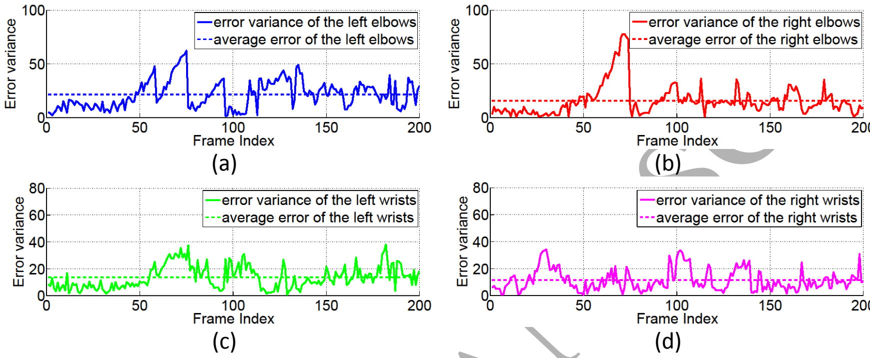


Figure 17: Pose error variance and average error of the joints of left/right elbows and left/right wrists.

Additionally, we also test our proposed method on a video file containing 200 frames to check the existence of drift while tracking. The pose error variance has been computed over entire sequence and is shown in Fig. 17. The obtained results clearly suggest that the error does not accumulate over time and hence, the method does not suffer from drift. Moreover, it is evident that the method is able to robustly converge on good poses in new frames following large errors in previous frames.

5.3. Visual Comparisons of Performance

Fig. 18 shows example visualisations of our method's results in comparison with the methods of [7] and [8] testing on the *VideoPose2.0-training* dataset, while Fig. 19 shows results for the *VideoPose2.0-testing* dataset. Fig. 20 shows results for the *Pose in the Wild* dataset. To compare with [7], we use the keypoints of our *coupled-layer DS puppet model* to draw stick poses, in order that poses are presented in the same way as [7]. In each comparison pair set, the first row represents the results of our method and the second row shows results for the comparison methods. Several instances can be seen where our method correctly estimates a pose while [7] and [8] generate substantial pose errors. Also check the provided supplementary video for better understanding of the results.

The second row of the first pair set in Fig. 18(a) shows that the person's lower arm jumps to a poor pose estimate (second and fourth columns), this problem is caused by a higher image likelihood of colour and contour when using Zuffi *et al.* 's method. In contrast, our proposed method overcomes this problem by exploiting an adaptive penalty term. The second row of the third pair set in Fig. 18(a) shows significant errors and erratic pose changes for Zuffi *et al.* . This is likely caused by the method of Zuffi *et al.* using a cost function for the entire body to evaluate each pose. In contrast, our proposed method evaluates the pose of each body part separately and then connects them according to a distance rule, which makes the resulting pose estimate more robust. The inaccuracy of Zuffi *et al.* in the second row of the third pair set in Fig. 19(a) is caused by the assumption that lower arms, in addition to hands, are always skin coloured. The second pair set in Fig. 20(a) illustrates the superiority of our method in calculating scale. When humans move from far to near ranges, our proposed method can robustly detect the scale change, whereas the method of [8] cannot.

The method of Cherian *et al.* requires a large quantity of human pose candidates, and then uses the the entire video sequence to mutually refine them. This method is able to improve the overall estimation accuracy level, but sacrifices making full use of the image likelihood of each frame.

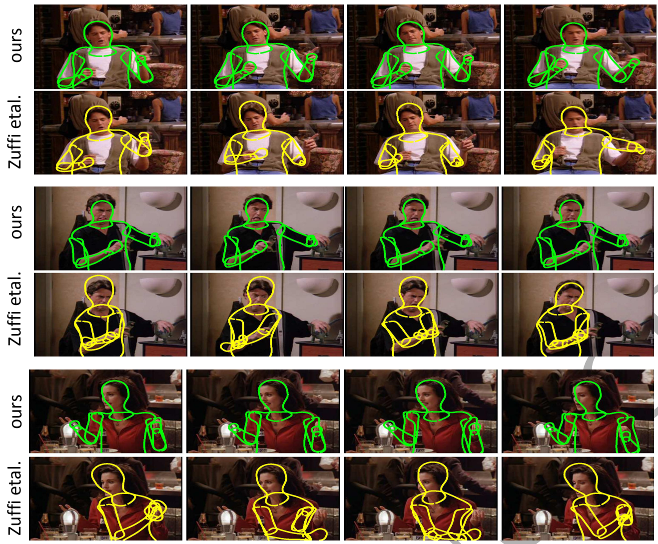
6. Conclusion

We have proposed a novel coupled-layer method for online human pose tracking, which demonstrates state-of-the-art adaptability, precision and robustness over a variety of video sequences. Global holistic models struggle to handle the

(a)

(b)

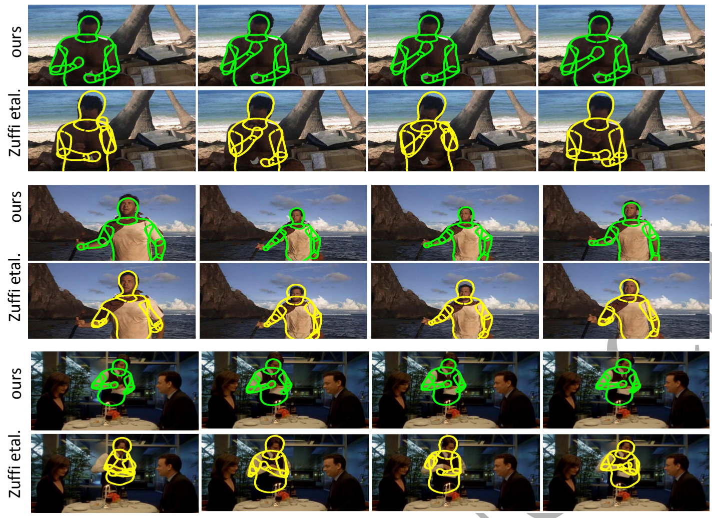
Figure 18: Example images comparing our results (using adaptive penalty) with the methods of Zuffi *et al.* [8] (sub-figure(a)) and Cherian *et al.* [7] (sub-figure(b)) on *VideoPose2.0-training* dataset. Each sub-figure has three pair sets, and in each pair set, the first row reveals sample results of our method, and the second row reveals the compared method.



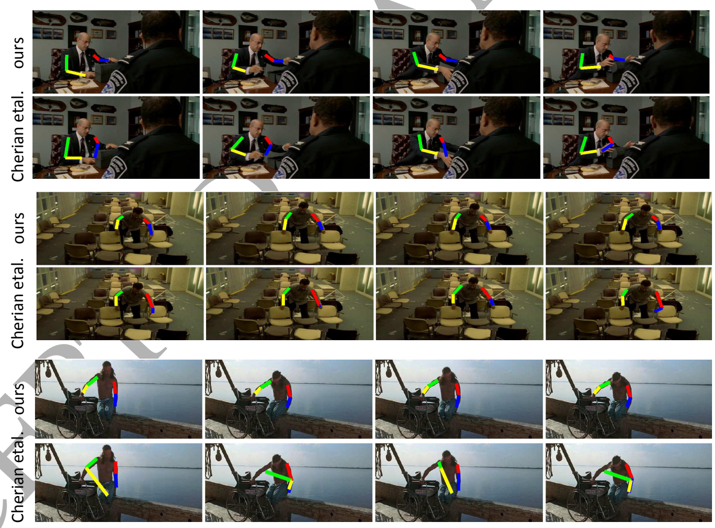
(a)

(b)

Figure 19: Sample results compared our results (using adaptive penalty) with the methods of Zuffi *et al.* [8] (sub-figure(a)) and Cherian *et al.* [7] (sub-figure(b)) on *VideoPose2.0-testing* dataset. Each sub-figure has three pair sets, and in each pair set, the first row reveals sample results of our method, and the second row reveals the compared method.



(a)



(b)

Figure 20: Sample results comparing our method (using adaptive penalty) with the methods of Zuffi *et al.* [8] (sub-figure(a)) and Cherian *et al.* [7] (sub-figure(b)) on *Pose in the Wild* dataset. Each sub-figure has three pair sets, and in each pair set, the first row reveals sample results of our method, and the second row reveals the compared method.

complexity of highly articulated objects, whereas parts-based methods lead to pose errors if not sufficiently constrained. Our coupled layer model combines elements of each approach to outperform previous methods. We also incorporated an adaptive motion penalty which can correct the pose of a human body part which has drifted from the previous frame. Our method relies only on the present and previous frames (except the first frame), and so is suitable for online sequential tracking. However, it still outperforms offline methods which rely on mutually optimising poses at all frames over the entire video sequence.

Acknowledgements

This work was funded in part by a scholarship from Shandong University, China, and has also been supported in part by the Innovate UK KTP partnership 9573 between the KUKA Robotics UK Ltd. and the University of Birmingham.

Authors would like to thank Dr. Silvia Zuffi and Dr. Anoop Cherian for making their code publicly available, and for providing us with valuable help and advice for validating and comparing our proposed method.

References

[1] D. Hogg, Model-based vision: a program to see a walking person, Image and Vision Computing. 1 (1) (1983) 5–20.

[2] A. Yao, J. Gall, G. Fanelli, L. J. Van Gool, Does human action recognition benefit from pose estimation, in: Proceedings of the British Machine Vision Conference. BMVC Press, Vol. 3, 2011, p. 6.

[3] X. P. Burgos-Artizzu, P. Dollár, D. Lin, D. J. Anderson, P. Perona, Social behavior recognition in continuous video, in: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 1322–1329.

[4] J. Charles, T. Pfister, D. Magee, D. Hogg, A. Zisserman, Domain adaptation for upper body pose tracking in signed tv broadcasts, in: Proceedings of the British Machine Vision Conference. BMVC Press, 2013.

[5] P. Agarwal, S. Kumar, J. Ryde, J. J. Corso, V. N. Krovi, Estimating human dynamics on-the-fly using monocular video for pose estimation, in: Robotics: Science and Systems, Citeseer, 2012.

[6] B. Sapp, D. Weiss, B. Taskar, Parsing human motion with stretchable models, in: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 1281–1288.

[7] A. Cherian, J. Mairal, K. Alahari, C. Schmid, Mixing body-part sequences for human pose estimation, in: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 2361–2368.

[8] S. Zuffi, J. Romero, C. Schmid, M. J. Black, Estimating human pose with flowing puppets, in: International Conference on Computer Vision (ICCV), IEEE, 2013, pp. 3312–3319.

[9] S. Zuffi, O. Freifeld, M. J. Black, From pictorial structures to deformable structures, in: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 3546–3553.

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

[11] M. Dantone, J. Gall, C. Leistner, L. Van Gool, Human pose estimation using body parts dependent joint regressors, in: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 3041–3048.

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

[13] L. Pishchulin, A. Jain, M. Andriluka, T. Thormahlen, B. Schiele, Articulated people detection and pose estimation: Reshaping the future, in: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 3178–3185.

[14] M. Eichner, M. Marin-Jimenez, A. Zisserman, V. Ferrari, 2d articulated human pose estimation and retrieval in (almost) unconstrained still images, International Journal of Computer Vision (IJCV). 99 (2) (2012) 190–214.

[15] D. Park, D. Ramanan, N-best maximal decoders for part models, in: International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 2627–2634.

[16] O. Freifeld, A. Weiss, S. Zuffi, M. J. Black, Contour people: A parameterized model of 2d articulated human shape, in: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 639–646.

[17] Y. Wu, T. S. Huang, Capturing articulated human hand motion: A divide-and-conquer approach, in: International Conference on Computer Vision (ICCV), Vol. 1, IEEE, 1999, pp. 606–611.

[18] K. Paul, M. Dimitrios, N. Jean-Christophe, Integration of bottom-up/top-down approaches for 2d pose estimation using probabilistic gaussian modelling, Computer Vision and Image Understanding (CVIU). 115 (2) (2011) 242–255.

[19] M. W. Lee, R. Nevatia, Human pose tracking using multi-level structured models, in: European Conference on Computer Vision (ECCV), Springer Berlin Heidelberg, 2006, pp. 368–381.

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

[21] D. Ramanan, D. A. Forsyth, A. Zisserman, Strike a pose: Tracking people by finding stylized poses, in: Computer Vision and Pattern Recognition (CVPR), Vol. 1, IEEE, 2005, pp. 271–278.

[22] V. Ferrari, M. Marín-Jiménez, A. Zisserman, 2d human pose estimation in tv shows, in: Statistical and Geometrical Approaches to Visual Motion Analysis, Springer, 2009, pp. 128–147.

[23] L. Sigal, M. J. Black, Measure locally, reason globally: Occlusion-sensitive articulated pose estimation, in: Computer Vision and Pattern Recognition (CVPR), Vol. 2, IEEE, 2006, pp. 2041–2048.

[24] V. I. Morariu, D. Harwood, L. S. Davis, Tracking people's hands and feet using mixed network and/or search, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). 35 (5) (2013) 1248–1262.

[25] K. Fragkiadaki, H. Hu, J. Shi, Pose from flow and flow from pose, in: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 2059–2066.

[26] N. G. Cho, A. L. Yuille, S. W. Lee, Adaptive occlusion state estimation for human pose tracking under self-occlusions, Pattern Recognition 46 (3) (2013) 649–661.

[27] X. Chen, A. L. Yuille, Articulated pose estimation by a graphical model with image dependent pairwise relations, in: Advances in Neural Information Processing Systems(NIPS), 2014, pp. 1736–1744.

[28] L. Sigal, M. J. Black, Predicting 3d people from 2d pictures, Articulated Motion and Deformable Objects. (2006) 185–195.

[29] Y. Yang, D. Ramanan, Articulated pose estimation with flexible mixtures-of-parts, in: Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 1385–1392.

[30] X. Burgos-Artizzu, D. Hall, P. Perona, P. Dollár, Merging pose estimates across space and time, in: Proceedings of the British Machine Vision Conference. BMVC Press, Citeseer, 2013.

[31] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, J. Davis, Scape: shape completion and animation of people, in: ACM Transactions on Graphics (TOG), Vol. 24, ACM, 2005, pp. 408–416.

[32] L. Cehovin, M. Kristan, A. Leonardis, Robust visual tracking using an adaptive coupled-layer visual model, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). 35 (4) (2013) 941–953.

[33] S. J. Pan, Q. Yang, A survey on transfer learning., IEEE Transactions on Knowledge and Data Engineering 22 (10) (2010) 1345–1359.

[34] P. Buehler, M. Everingham, D. P. Huttenlocher, A. Zisserman, Upper body detection and tracking in extended signing sequences, International Journal of Computer Vision (IJCV). 95 (2) (2011) 180–197.

[35] C. Liu, Beyond pixels: exploring new representations and applications for motion analysis, Ph.D. thesis, Massachusetts Institute of Technology (2009).

[36] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, Advances in large margin classifiers 10 (3) (1999) 61–74.