Pose-Guided Human Parsing with Deep-Learned Features

Fangting Xia, Jun Zhu*, Peng Wang*, Alan Yuille
University of California, Los Angeles

Abstract

Parsing human body into semantic regions is crucial to human-centric analysis. In this paper, we propose a segment-based parsing pipeline that explores human pose information, i.e. the joint location of a human model, which improves the part proposal, accelerates the inference and regularizes the parsing process at the same time. Specifically, we first generate part segment proposals with respect to human joints predicted by a deep model [5], then part-specific ranking models are trained for segment selection using both pose-based features and deep-learned part potential features. Finally, the best ensemble of the proposed part segments are inferred though an And-Or Graph. We evaluate our approach on the popular Penn-Fudan pedestrian parsing dataset [27], and demonstrate the effectiveness of using the pose information for each stage of the parsing pipeline. Finally, we show that our approach yields superior part segmentation accuracy comparing to the state-of-the-art methods.

1. Introduction

The goal of human parsing is to partition the human body into different semantic parts such as hair, head, torso, arms, legs, etc, which provides rich descriptions for human-centric analysis, and thus becomes increasingly important to many computer vision applications, including content-based image/video retrieval [30, 24], person re-identification [19, 6], video surveillance [24, 34, 18], action recognition [29, 25, 37] and clothes fashion recognition [32]. However, it is very challenging in real-life scenarios due to variability in human appearances and shapes, caused by large numbers of human poses, clothes types, and occlusion/self-occlusion patterns.

Current state-of-the-art approach for human parsing is the segment-based graphical model framework by first generating segment/region proposals for parts based on appearance similarity, then selecting and assembling the segments by a graphical model [2, 35, 7, 16]. However, using only

---

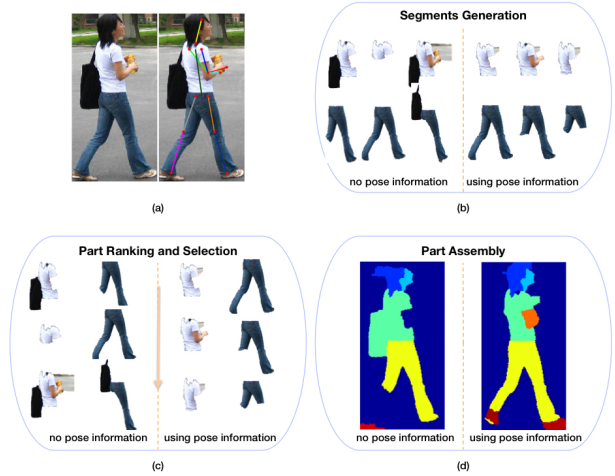* indicates equal contributions.

Figure 1. Human parsing using effective pose cues and deep cues. (a) Left: original image. Right: predicted pose. (b) Pose-based part proposal. Left panel: without pose information. Right panel: with pose information. (c) Part ranking and selection. Left panel: leading part proposals without pose cues. Right panel: leading part proposals with pose cues. (d) Final parsing results. Left panel: without pose cues. Right panel: with pose cues.

bottom-up cues has difficulties in locating good part segments with color ambiguities or noisy regions. Previous strategy [7] integrated pose cues to handle the problem, while the pose model is only used at last stage, where the error made from part proposals is inevitably propagated. As illustrated in Fig. 1 and Fig. 2, our approach synergies the accurate top-down pose cues in all the parsing process with deep learned features, which largely improves the quality of the part proposal (in Fig.1(b)), provides robust feature for ranking (in Fig.1(c)) and regularizes the graphical ensemble (in Fig.1(d)).

The framework of our approach is illustrated in Fig. 2. Given an image, firstly, as shown at the bottom, the pose information of the human inside is estimated using a deep pose approach [5] as the overall parsing guidance in each stage. Our estimation can be very accurate due to the human poses in our parsing task is in a constraint situation, i.e. mostly walking and standing.

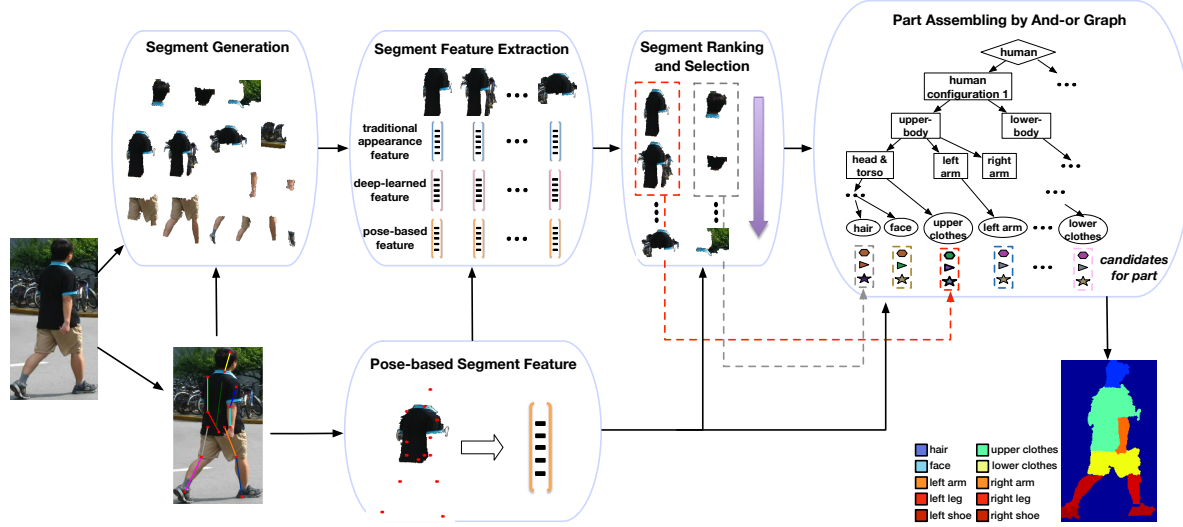At the top row, we show the three stages of our parsing

Figure 2. Illustration of our human parsing pipeline.

pipeline. In the first stage, a pool of part segment proposals are generated with the information that parts should appear around pose joints, yielding proposals with high precision and recall. In the second stage, rich feature description including pose features and appearance features are proposed to describe each segment proposal, based on which a regressor is trained to re-rank the proposals. Specifically, the pose feature captures spatial relationship between a proposed segment and the predicted human pose joints, both locally and globally.The appearance feature is computed from both hand-designed features and the deep-learned part potential using the fully convolutional network (FCN) [17], which models a part proposal's shape and appearance. The two top-down features are complementary and provide robust cues to prune false positive proposals from the background. After pruning the proposals, a small number of high-quality segment proposals for each part category are selected for the part assembling stage. In the third stage, an And-Or graph (AOG) [40, 39, 38, 7] is applied to optimally assemble the selected proposals of different parts into the final parsing result, in which the pose cue is additionally utilized to measure the pairwise context compatibility. We evaluate our method on the popular *Penn-Fudan* [27] pedestrian parsing benchmark, and show that the pose information effectively improve the performance in every stage of our parsing pipeline, and by incorporating the deep-learned potential features, our approach outperforms other state-of-the-arts in this human parsing task with a significant margin.

In summary, the contributions of this work are in three folds:

(1) We develop a human parsing pipeline that systematically explores the top-down pose information at every stage to regularize the model, yielding strong improvements in parsing efficiency and effectiveness w.r.t. the

state-of-the-art methods.

(2) We effectively incorporate the deep learned features for each part proposal, which provide robust representation of part appearance.

(3) We propose a novel pose-based geometric feature that models the spatial relationship of different segments and parts, which is substantially important to part selection and composition.

2. Related Work

In the literature of human parsing, the generation, assembling framework produces the state-of-the-art results. We will first review these parsing methods w.r.t the two stages.

Part segment proposal generation. Previous works [2, 32, 35] usually adopt low-level segment-based proposal. For example, [32, 35] use uniform appearance superpixels as the elemental proposals of body parts. Some approaches take higher level cues. Bo and Fowlkes [2] exploited roughly learned part location priors and part mean shapes information, and derive a number of part segments from the gPb-UCM method [1] using a constrained region merging method. Dong *et al.* [7] employed the Parselets [8] for proposal to obtain mid-level part semantic information for proposal. However, either low-level, mid-level or rough location proposals may result in many false positives, misleading the later process. In our approach, we embed the top-down accurate pose joint cues directly into the efficient bottom-up generation algorithm [10] to generate "pose-guided" proposals, which significantly avoids many false positives and improves the segment quality.

Part assembling. Given the generated part segments, an assemble model takes in the selected part segments and outputs the final results, where part unary potentials and rela-

tive relationships are leveraged in the model. Bo *et al.* [2] developed a compositional model that model human parsing into two different levels of body parts. It uses a series of hard geometric constraints, e.g., face and hair should be adjacent, to model relative part geometry in the inference process. In [32, 35], a conditional random field (CRF) is built on top of the superpixels to label part categories. In [7], a hybrid parsing model (HPM) is proposed to integrate human part parsing and pose estimation, yielding consistent results in both tasks. In these works, pose information was used in [32] and [7] to improve the results. However, our method differs and improves from previous works in the following several aspects: (1) Rather than hand crafted features used in [7, 32], we use the deep learned features both for pose and appearance, yielding more robust representations. (2) Our pose feature descriptor measures consistency between pose and segments both locally and globally. (3) Most importantly, the pose information is embedded systematically in the whole pipeline including the part segment proposal, part selection and assembling, yielding more efficient inference and more robust human body parsing.

Most recently, some studies also try to adopt deep features in this tasks, yielding impressive results. Luo *et al.* [18] proposed a deep decompositional network (DDN) to parse pedestrian images into semantic regions, which uses pixel-wise HOG feature map as the input of their system. Liu *et al.* [16] adopted non-parametric methods and trained a matching deep network to match the input image region to the retrieved ones. In [9, 23], they applied the FCN [17] to solve human/object parsing in an end-to-end manner. Wang *et al.* [28] extended it to a two-channel FCN that jointly tackles object segmentation and part segmentation on some animal classes. However, due to lack of the pose to regularize the parsing model, some false positives can not be effectively avoided. Our work gives a first attempt to embed the pose cue with the deep learned parsing strategies, showing that it is an important complementary information.

3. The pose-guided human parsing pipeline

Given a pedestrian image I, we first adopt the existing state-of-the-art pose estimation approach [5] to predict a series of human pose joints $\mathcal{L} = \{l_1, l_2, \cdots, l_{n_l}\}$, where l_j denotes the location of the j-th pose joint, and $n_l = 14$ in this paper. Here we use the same joints as those commonly used in the human pose estimation literature [36, 5], i.e. forehead, neck, shoulders, elbows, wrists, hips, knees, and ankles. Based on the human pose joint cues, our human parsing pipeline has three successive steps: *part segment proposal generation*, *part proposal selection*, and *part assembling*, each of which leverages the pose information as shown in Fig. 2. We will elaborate on the three steps in the following subsections respectively.

3.1. Pose-guided part segment proposal generation

To generate part segment proposals, we adopt the RIGOR segment proposal method [10] which is based on the min-cut algorithm [3] and can efficiently generate segments aligning with object boundaries given user defined initial seeds and cutting thresholds. In our scenario, we generate the seeds based on the predicted pose joint locations. Specifically, given the observation that the part segments tend to be surrounding the corresponding pose joints, for each joint j we sample a set of seeds at the 5×5 grid locations over a 40×40 image patch centered at this joint. We use 8 different thresholds, yielding 200 segment proposals in total for each joint.

Further, we prune out duplicate or highly similar segments and construct a segment proposal pool $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$, where the segments are unique to each other. In detail, we sequentially add the generated segment proposal only if the intersection over union (IoU) value w.r.t. each existing segment is less than 0.95. Finally we generate a pool of around 800 segment proposals for each image, and use them as candidate part segments in latter two steps.

3.2. Part proposal selection

Directly feeding a number of segments into the part assembling model leads to very high computational cost due to the existence of pairwise or high-order terms in the AOG model. Thus, for each part we present a proposal selection step to prune the segments with low probability of being that part class, resulting in much less candidate segments for the part assembling step.

Specially, for each segment proposal $s_i \in \mathcal{S}$, we consider multiple features from a variety of cues on appearance, shapes and poses shown as below:

- $\phi^{o2p}(s_i)$, a second order pooling (O2P) feature [4] for describing appearance cues.

- $\phi^{skin}(s_i)$, an appearance feature capturing skin color cues. We adopt the method of [13] to produce a skin potential for each pixel in s_i, and $\phi^{skin}(s_i)$ is computed via the second order pooling operation on the skin potential map of s_i.

- $\phi^{pbg}(s_i, \mathcal{L})$, a posed-based geometric (PBG) feature we proposed in this paper, which measures the spatial relationship between the segment s_i and the predicted pose joint configuration $\mathcal{L}$. We will elaborate on this feature in Sec. 4.1.

- $\phi^{c-pbg}(s_i, \mathcal{L})$, a coded posed-based geometric (C-PBG) feature which is computed as the an encoded version of $\phi^{pbg}(s_i, \mathcal{L})$ using a dictionary. It linearizes the feature space of ϕ^{pbg}, and facilitates to learn the linear regressor later. The details will be given in Sec. 4.2.
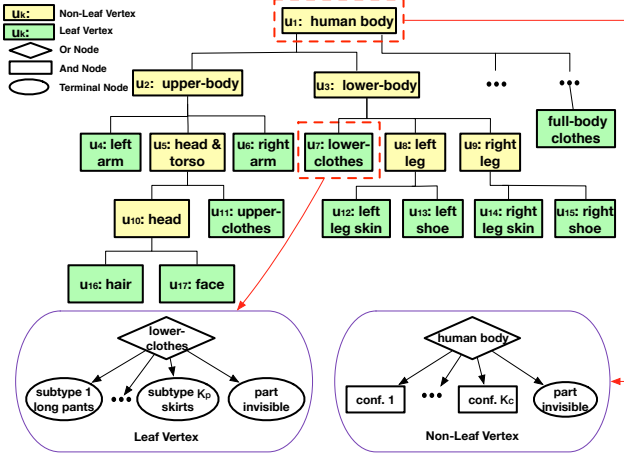
Figure 3. Illustration on the architecture of our AOG model.

- $\phi^{fcn}(s_i, \mathcal{H})$, a feature computed from the deep-learned potential maps $\mathcal{H}$ using FCN [17]. It measures the compatibility between the low-level segment image patch and high-level part semantic cues from FCN, and we will introduce the details in Sec. 4.3.

Our final feature descriptor of s_i is the concatenation of the aforementioned features, i.e.,

$$\phi(s_i, \mathcal{L}, \mathcal{H}) = [\phi^{o2p}(s_i), \phi^{skin}(s_i), \phi^{fcn}(s_i, \mathcal{H}),$$
$$\phi^{pbg}(s_i, \mathcal{L}), \phi^{c-pbg}(s_i, \mathcal{L})]^{\mathrm{T}} \quad (1)$$

On basis of this hybrid feature representation, we train a linear support vector regressor (SVR) [4] for each part category. Let P denote the total number of part categories and $p \in \{1, 2, \cdots, P\}$ denote the index of a part category. The target variable for training SVR is the IoU value between the segment proposal and ground-truth label map of part category p. The output of SVR model is given by Equ. (2).

$$g^p(s_i|\mathcal{L}, \mathcal{H}) = \beta_p^{\mathrm{T}} \phi(s_i, \mathcal{L}, \mathcal{H}), \quad (2)$$

where β_p is the model parameter of SVR for the p-th part category. Thus, for any part category p, we rank the segment proposals in $\mathcal{S}$ based on their SVR scores $\{g^p(s_i) \,|\, s_i \in \mathcal{S}\}$. Finally, we select the top-n_p scored segments separately for each part category and combine the selected segment proposals from all part categories to form a new segment pool $\tilde{\mathcal{S}} \subseteq \mathcal{S}$. In this paper, we set $n_p = 10$ such that the number of selected segments in $\tilde{\mathcal{S}}$ is much smaller than N.

3.3. Part assembling with And-Or Graph

There are two different groups of classes (i.e., the *parts* and the *part compositions*) in our AOG model: the part classes are the finest-level constituents of human body, e.g. face, arms, legs, etc.; the part composition classes correspond to intermediate concepts in the hierarchy of semantic human body constituents, e.g. head, upper-body, lower-body, etc., each of which consists of multiple parts. We list

Table 1. The full list of parts and part compositions in our AOG.

Part	Part Composition
hair face	head
full-body clothes	head & torso
upper-clothes	upper-body
left/right arm	lower-body
lower-clothes	left/right leg
left/right leg skin	human body
left/right shoe	

all the parts and part compositions in table 1. To assemble the selected part segments, we develop a compositional AOG model as illustrated in Fig. 3, which facilitates flexible composition structure and standard learning/inference routines. Let C denote the total number of part compositions. Formally, our AOG model is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \mathcal{T} \cup \mathcal{N}$ denotes a set of vertices and $\mathcal{E}$ refers to the set of edges associated. Meanwhile, $\mathcal{T} = \{1, 2, \cdots, P\}$ and $\mathcal{N} = \{P+1, P+2, \cdots, P+C\}$ denote the set of part indices and the set of part composition indices respectively. In our AOG, each leaf vertex $p \in \mathcal{T}$ represents one human body part and each non-leaf vertex $c \in \mathcal{N}$ represents one part composition. The root vertex corresponds to the whole human body while the vertices below correspond to the part compositions or parts at various semantic levels. Our goal is to parse the human body into a series of part compositions and parts, which is in a hierarchical graph instantiated from the AOG model.

The vertex of our AOG is a nested subgraph as illustrated at the bottom in Fig. 3. For a leaf vertex $p \in \mathcal{T}$, it includes one Or-node followed by a set of terminal nodes as its children. The terminal nodes correspond to different part types, and the Or-node represents a mixture model indicating the selection of one part type from terminal nodes. Formally, we define a state variable $z_p \in \{0, 1, 2, \cdots, K_p\}$ to indicate that the Or-node selects the z_p-th terminal node as the part type for leaf vertex p. As the example of green node in Fig. 3, the lower-clothes part can select one kind of type (e.g. long pants or skirt) from its candidate part types. Besides there is one special terminal node representing the invisibility of this part due to occlusion/self-occlusion, which corresponds to the state $z_p = 0$. For a non-leaf vertex $c \in \mathcal{N}$, it includes one Or-node linked by a set of And-nodes plus one terminal node. The Or-node of non-leaf vertex represents this part composition has several different ways of decompositions into smaller parts and/or part compositions. The And-node corresponds to one alternative configuration of decomposition for c. As shown in Fig. 3, the non-leaf vertex head can be composed by one of several different configurations of two child vertices (i.e., face and hair). Similar to the leaf vertices, we also induce
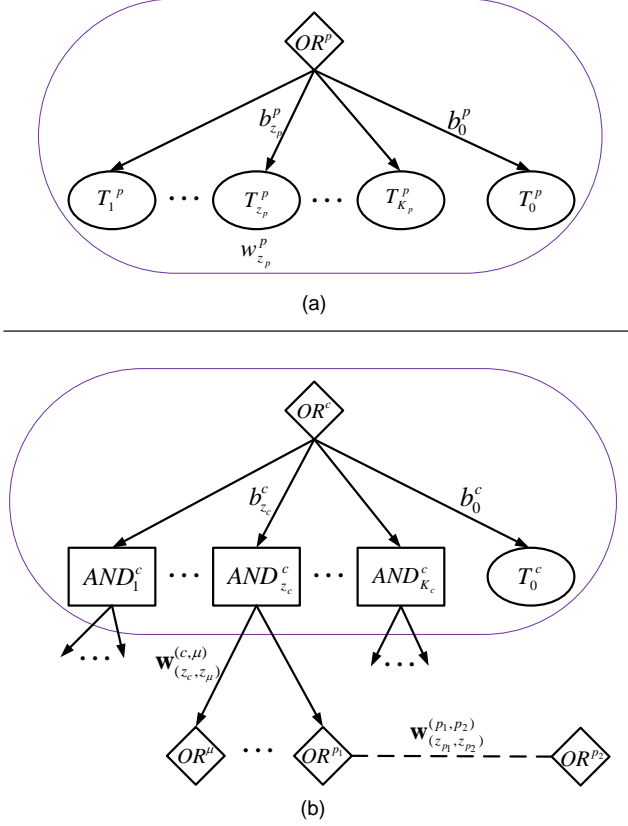
(a)

(b)

Figure 4. Illustration on the structure of vertices in AOG. (a) leaf vertex; (b) non-leaf vertex. The symbols of *OR*, *AND* and *T* represent the Or-node, And-node and terminal node respectively. Please see Eqn. (5) and Eqn. (6) about the notations of model parameters.

a state variable $z_c \in \{0, 1, 2, \cdots, K_c\}$ to indicate that the Or-node of part composition c selects the z_c-th And-node as the configuration of child vertices for $z_c \neq 0$ or this part composition is invisible when $z_c = 0$.

Furthermore, we define another state variable y to indicate the selection of segment from the candidate pool of a part or part composition. For a leaf vertex $p \in \mathcal{T}$, $y_p \in \{0, 1, 2, \cdots, n_{p,z_p}\}$ represents that the part p selects the y_p-th segment proposal (i.e., $s_{y_p}^{p,z_p}$) from the segment pool $\tilde{S}_{p,z_p}$ output by its segment ranking model on type z_p. Meanwhile, $y_p = 0$ is a special state which coincides with the invisibility pattern of part p (i.e., $z_p = 0$). To make notation consistent, we use s_0^{p,z_p} to represent an "null" segment for part invisibility. For a non-leaf vertex $c \in \mathcal{N}$, $y_c \in \{0, 1, 2, \cdots, n_c\}$ indicates a segment $s_{y_c}^{c,z_c} \in \tilde{S}_{c,z_c}$ is selected, where $s_{y_c}^{c,z_c}$ is obtained by the union of its child vertices' candidate segments and $\tilde{S}_{c,z_c}$ denotes the candidate segment pool for the z_c And-node. When $y_c = 0$, likewise, the s_0^{c,z_c} represents a null segment indicating the invisibility pattern of part composition c. Let $Ch(c, z_c)$ denote the set of child vertices for part composition c and configuration z_c.

Formally, $s_{y_c}^{c,z_c}$ is defined by Equ. (3).

$$s_{y_c}^{c,z_c} = \mathbb{U}_{\mu \in Ch(c,z_c)} s_{y_\mu}^{\mu,z_\mu}, \tag{3}$$

where $\mathbb{U}$ represents a pixel-wise union operation of combing the child vertices' segment masks to generate a new segment.

Let $\mathbf{Y} = (y_1, y_2, \cdots, y_P, y_{P+1}, y_{P+2}, \cdots, y_{P+C})$ and $\mathbf{Z} = (z_1, z_2, \cdots, z_P, z_{P+1}, z_{P+2}, \cdots, z_{P+C})$ denote the structural solution of AOG. We define a global score function of AOG $F(\mathbf{Y}, \mathbf{Z} | \tilde{S}, \mathcal{L}, \mathcal{H})$[1] to measure the compatibility between $(\mathbf{Y}, \mathbf{Z})$ and $(\tilde{S}, \mathcal{L}, \mathcal{H})$ for image I, which can be calculated as shown in Equ. (4).

$$F(\mathbf{Y}, \mathbf{Z}|\tilde{S}, \mathcal{L}, \mathcal{H}) = \sum_{p \in \mathcal{T}} f(y_p, z_p) \tag{4}$$
$$+ \sum_{c \in \mathcal{N}} f(y_c, z_c, \{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\}),$$

where $f(y_p, z_p)$ is a local scoring function of leaf vertex p and $f(y_c, z_c, \{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\})$ denotes a scoring function of the partial graph rooted by non-leaf vertex c. For each leaf vertex $p \in \mathcal{T}$ (i.e., a part), we compute $f(y_p, z_p)$ by Equ. (5).

$$f(y_p, z_p) = \begin{cases} b_{z_p}^p + w_{z_p}^p \cdot g_{z_p}^p(s_{y_p}^{p,z_p}|\mathcal{L}, \mathcal{H}), & z_p \neq 0 \tag{5} \\ b_0^p, & z_p = 0 \end{cases}$$

where $w_{z_p}^p$ and $b_{z_p}^p$ denote the weight and bias parameters of unary term for part p respectively. Particularly, b_0^p is the bias parameter for the invisibility pattern of p. Besides, $g_{z_p}^p$ is dependent on the part type z_p, implying the regression models defined in Equ. (2) are trained by different parts and types. Fig. 4 (a) illustrates the structure of a leaf vertex as well as corresponding model parameters.

For each non-leaf vertex $c \in \mathcal{N}$ (i.e., a part composition), we compute $f(y_c, z_c, \{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\})$ by Eqn. (6).

$$f(y_c, z_c, \{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\}) \tag{6}$$
$$= \begin{cases} b_{z_c}^c + u(y_c, z_c, \{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\}), & z_c \neq 0 \\ b_0^c, & z_c = 0 \end{cases}$$

where

$$u(y_c, z_c, \{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\}) \tag{7}$$
$$= h(\{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\})$$
$$+ \sum_{\mu \in Ch(c,z_c)} \mathbf{w}_{(z_c, z_\mu)}^{(c,\mu) \ \mathrm{T}} \varphi(s_{y_c}^{c,z_c}, s_{y_\mu}^{\mu,z_\mu})$$
$$+ \sum_{(p_1,p_2) \in \mathcal{R}_c} \mathbf{w}_{(z_{p_1}, z_{p_2})}^{(p_1,p_2) \ \mathrm{T}} \psi(s_{y_{p_1}}^{p_1,z_{p_1}}, s_{y_{p_2}}^{p_2,z_{p_2}}|\mathcal{L}),$$

---

[1] Here $\tilde{S} = \bigcup_{p \in \mathcal{T}, z_p \neq 0} \tilde{S}_{p,z_p}$

Table 2. The list of adjacent part pairs.

Part Composition (c)	Adjacent Part Pairs ($\mathcal{R}_c$)
human body	(upper-clothes, lower-clothes), (full-body clothes, left leg skin), (full-body clothes, right leg skin)
head	(hair, face)
head & torso	(upper-clothes, hair), (upper-clothes, face), (full-body clothes, hair), (full-body clothes, face)
upper-body	(left arm, upper-clothes), (right arm, upper-clothes)
	(left arm, full-body clothes), (right arm, full-body clothes)
lower-body	(lower-clothes, left leg skin), (lower-clothes, right leg skin)
	(lower-clothes, left shoe), (lower-clothes, right shoe)
left leg	(left leg skin, left shoe)
right leg	(right leg skin, right shoe)

and

$$h(\{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\}) = \sum_{\mu \in Ch(c, z_c) \cap \mathcal{T}} f(y_\mu, z_\mu) \quad (8)$$
$$+ \sum_{\mu \in Ch(c, z_c) \cap \mathcal{N}} f(y_\mu, z_\mu, \{(y_\nu, z_\nu) : \nu \in Ch(\mu, z_\mu)\}).$$

Concretely, Eqn. (6) can be divided into four terms:

(1) the bias term of selecting z_c for the Or-node, i.e. $b_{z_c}^c$. b_0^c is the bias parameter when part composition c is invisible (In this case, all the descendant vertices are also invisible and thus the latter three terms are zero).

(2) the sum of scores of its child vertices for the selected And-node, i.e. $h(\{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\})$.

(3) the sum of parent-child pairwise terms (i.e., *vertical edges*) for measuring the spatial compatibility between the segment of part composition c and the segments of its child vertices, i.e. $\sum_{\mu \in Ch(c, z_c)} \mathbf{w}_{(z_c, z_\mu)}^{(c,\mu)}{}^{\mathrm{T}} \varphi(s_{y_c}^{c, z_c}, s_{y_\mu}^{\mu, z_\mu})$, where $\varphi(s_{y_c}^{c, z_c}, s_{y_\mu}^{\mu, z_\mu})$ denotes a spatial compatibility feature of segment pair $(s_{y_c}^{c, z_c}, s_{y_\mu}^{\mu, z_\mu})$ and $\mathbf{w}_{(z_c, z_\mu)}^{(c,\mu)}$ refers to corresponding weight vector. Specifically, φ is defined by $[dx; dx^2; dy; dy^2; ds; ds^2]$, in which dx, dy represent the spatial displacement between the center locations of two segments while ds is the scale ratio of them.

(4) the sum of pairwise terms (i.e., *side-way edges*) for measuring the geometric compatibility on all segment pairs specified by an adjacent part-pair set $\mathcal{R}_c$, which defines a couple of adjacent part pairs for c (e.g., for the part composition of lower body, we consider lower-clothes and leg skin to be an adjacent part pair)[2]. To avoid double counting in recursive computation of Eqn. (7), $\mathcal{R}_c$ only includes the relevant part pairs which have at least one child vertex of c. This side-way pairwise potential corresponds to

---
[2] We list the adjacent part pairs for each part composition in table 2.

$\sum_{(p_1, p_2) \in \mathcal{R}_c} \mathbf{w}_{(z_{p_1}, z_{p_2})}^{(p_1, p_2)}{}^{\mathrm{T}} \psi(s_{y_{p_1}}^{p_1, z_{p_1}}, s_{y_{p_2}}^{p_2, z_{p_2}} | \mathcal{L})$ in Eqn. (7), where $\psi(s_{y_{p_1}}^{p_1, z_{p_1}}, s_{y_{p_2}}^{p_2, z_{p_2}} | \mathcal{L})$ represents a geometric compatibility feature of segment pair $(s_{y_{p_1}}^{p_1, z_{p_1}}, s_{y_{p_2}}^{p_2, z_{p_2}})$ and $\mathbf{w}_{(z_{p_1}, z_{p_2})}^{(p_1, p_2)}$ is corresponding weight vector. In this paper, we use the coded version of joint-segment geometric feature for ψ, which will be elaborated in Sec. 4.2.

In Fig. 4 (b), we illustrate the structure of a non-leaf vertex and corresponding model parameters.

According to the hierarchical architecture of our AOG, $F(\mathbf{Y}, \mathbf{Z} | \mathcal{L}, \tilde{\mathcal{S}}, \mathcal{H})$ can be recursively computed by Eqn. (5) and Eqn. (6) in a bottom-up manner. In practice, we first calculate the score for each leaf vertex, and then calculate the scores of non-leaf vertices from the lowest levels to the root part composition vertex (i.e., human). It is noted that our AOG is not a directed acyclic graph due to the existence of side-way edges, which makes loops among a clique of some vertices. This prevent our model from using common dynamic programming for inference. In this paper, we propose a greedy pruning algorithm for model inference and will elaborate on it in Sec. 5.2.

4. Feature design

In this section, we elaborate on the aforementioned features in part proposal selection (Sec. 3.2) and part assembling (Sec. 3.3) steps.

4.1. The pose-based geometric (PBG) feature

The PBG feature of a segment $\phi^{pbg}(s_i, \mathcal{L})$ is calculated based on the spatial relationship between the segment s_i and the predicted pose joints $\mathcal{L}$. As shown in Fig. 5, centered at s_i, the image is equally divided into eight orientations (I − VIII) and three region scales (S1, S2 and S3), yielding 24 spatial bins in total. Then each joint $l_j \in \mathcal{L}$ falls into one of these spatial bins and produces a 24 dimensional binary feature, quantizing the spatial relationship of l_j w.r.t. s_i. After that, we concatenate the binary features of all joints, and produce a $24 \times 14 = 336$ dimensional binary feature

Figure 5. Illustration of the pose-based geometric feature (PBG).

Figure 6. The learned prototypes/clusters for part category face. We show exemplar images for 3 out of 6 clusters. Cluster (1): frontal face or back face. Cluster (2): frontal/back face on the left. Cluster (3): side face on the left. The other clusters correspond to the symmetric patterns w.r.t. those shown here.

vector to describe the spatial relationship of segment s_i w.r.t. the pose joint configuration $\mathcal{L}$.

Specifically, S1 and S2 are the regions eroded and dilated by 10 pixels from the segments boundary respectively. S3 is the rest region of image. This segment-dependent definition of region scales depicts meaningful geometric cues from the predicted pose information. Intuitively, S1 constrains the involved joints inside the segment, while S2 implies the joints tend to be around the segment boundary. S3 indicates that the joints should be totally outside the segment. Thus, with our PBG feature, we can learn a more discriminative model by leveraging the geometric compatibility between the segments and pose joints.

4.2. The coded PBG feature (C-PBG)

The proposed pose feature can be highly non-linear in the feature space which might be suboptimal for linear classification. This motivates us to encode the pose feature though feature coding to achieve linearity, which has long been proofed to be effective in many vision tasks such as image classification [14, 33, 26, 15], semantic segmentation [4] etc. In this paper, we adopt a simple soft-assignment quantization (SAQ) coding method [15] to encode the PBG feature aforementioned.

At first, we collect the ground-truth segments from training images and compute their PBG features. After that, a dictionary of pose-guided part prototypes $\mathcal{D} = \{\mathbf{b}_m\}_{m=1}^{N_{\mathcal{D}}}$ can be learned via K-means clustering algorithm on the PBG feature representation of segment examples. To balance different part categories, we separately perform clustering and obtain $N_p = 6$ for each part category, resulting in a dictionary of $N_{\mathcal{D}} = K \times N_p$ codewords. Given $\mathcal{D}$, we compute the Euclid distance between original PBG feature of s_i and each prototype $\mathbf{b}_m$: $d_{i,m} =\parallel \phi^{pbg}(s_i, \mathcal{L}) - \mathbf{b}_m \parallel$. Thus the coded posed-based geometric (C-PBG) feature is formally defined as the concatenation of both the normalized and unnormalized codes w.r.t. $\mathcal{D}$:

$$\phi^{c-pbg}(s_i, \mathcal{L} \,|\, \mathcal{D}) = [a_{i,1}, \cdots, a_{i,N_{\mathcal{D}}}, a'_{i,1}, \cdots, a'_{i,N_{\mathcal{D}}}]^{\mathrm{T}}, \quad (9)$$

where $a_{i,m} = \exp^{(-\lambda d_{i,m})}$ and $a'_{i,m} = \frac{a_{i,m}}{\sum_{j=1}^{N_{\mathcal{D}}} a_{i,j}}$ denote the unnormalized and normalized code values w.r.t. $\mathbf{b}_m$ respectively. λ is a hyper-parameter of our coding method. As introduced in Sec. 3.2, the C-PBG feature is adopted in training the SVR models for part proposal selection. Besides, it is used for generating candidate part types in part assembling step. As illustrated in Fig. 6, the learned prototypes/clusters generally correspond to different typical views of the face part category.

In addition, we propose to code the pairwise PBG feature, i.e. $\psi(s_{y_{p_1}}^{p_1, z_{p_1}}, s_{y_{p_2}}^{p_2, z_{p_2}} \,|\, \mathcal{L})$ applied in Eqn. (6), for describing the geometric relationship between two adjacent parts p_1 and p_2. Specifically, we adopt the same coding process as before but using the concatenated PBG features of a pair of candidate segments $(s_{y_{p_1}}^{p_1, z_{p_1}}, s_{y_{p_2}}^{p_2, z_{p_2}})$. We perform clustering separately for each adjacent part pair and learn a class-specific dictionary for this pairwise C-PBG feature. In this paper, the dictionary size is set by $N_{pp} = 8$ for each part pair. As visualized in Fig. 7, the learned part-pair prototypes are very meaningful which capture typical view point and part type co-occurrence patterns for adjacent parts.

4.3. Deep-learned potential

As applied in [9, 28, 23], deep potentials for part, although lack of ability of discovering small parts and global pose regularization, provide strong high-level semantic evidence of part class and shape cues. Thus, we also combine this potential into the feature representation of segment proposals. Specifically, we train a FCN with 16s as in [28] with the output variables to be the part ground truth map.

Figure 7. The learned prototypes/clusters for the adjacent part pair *upper-clothes and lower-clothes*. We show 3 out of 8 clusters. Cluster (1): the person with short sleeved upper-clothes and short pants. Cluster (2): the person with short sleeved upper-clothes and long pants. Cluster (3): the person with long sleeved upper-clothes and long pants.

Given an image, it produces $P + 1$[3] potential maps $\mathcal{H}$, from which we also obtain $P + 1$ binary part class label masks $\mathcal{B}$ though MAP over each potential map. Thus, for a segment s_i, the deep feature $\phi^{fcn}(s_i, \mathcal{H})$ consists of three components: (1) the mean value inside s_i of all maps in $\mathcal{H}$; (2) the mean value along the contour of s_i from the maps in $\mathcal{H}$; (3) The IoU value between s_i to all the $K + 1$ masks in $\mathcal{B}$, i.e. $IoU(s_i, \mathcal{B})$.

5. Learning and inference for AOG

In this section, we first introduce the learning algorithm on our AOG model, and then elaborate on a greedy pruning algorithm for efficient inference.

5.1. Structural max-margin learning

The scoring function of Equ. (4) is a generalized linear model w.r.t. its parameters. Actually, we can concatenate all the model parameters to be a single vector $\mathbf{W}$ and rewrite Equ. (4) by $F(\mathbf{Y}, \mathbf{Z}|\mathcal{L}, \tilde{\mathcal{S}}, \mathcal{H}) = \mathbf{W}^T\boldsymbol{\Phi}(\mathcal{L}, \tilde{\mathcal{S}}, \mathcal{H}, \mathbf{Y}, \mathbf{Z})$. $\boldsymbol{\Phi}(\mathcal{L}, \tilde{\mathcal{S}}, \mathcal{H}, \mathbf{Y}, \mathbf{Z})$ is a re-organized sparse vector gathering all the features based on the structural state variable $(\mathbf{Y}, \mathbf{Z})$. In our AOG model, $\mathbf{Z}$ determines the topological structure of a feasible solution (i.e., parse tree), and $\mathbf{Y}$ specifies the segments selected for the vertices of this parse tree. Given a set of labelled examples $\{(\mathbf{Y}_n, \mathbf{Z}_n)|n = 1, 2, \cdots, J\}$, we formulate a structural max-margin learning problem on $\mathbf{W}$,

---

[3]The additional 1 corresponds to the background class.

as shown in Equ. (10).

$$\min_{\mathbf{W}} \frac{1}{2}\mathbf{W}^T\mathbf{W} + C \sum_{n=1}^{J} \xi_n \qquad (10)$$

$$\mathbf{W}^T\boldsymbol{\Phi}(\mathcal{L}_n, \tilde{\mathcal{S}}_n, \mathcal{H}_n, \mathbf{Y}_n, \mathbf{Z}_n) - \mathbf{W}^T\boldsymbol{\Phi}(\mathcal{L}_n, \tilde{\mathcal{S}}_n, \mathcal{H}_n, \mathbf{Y}, \mathbf{Z})$$

$$\geq \Delta(\mathbf{Y}_n, \mathbf{Z}_n, \mathbf{Y}, \mathbf{Z}) - \xi_n, \quad s.t. \ \forall \ \mathbf{Y} \text{ and } \mathbf{Z},$$

where $\Delta(\mathbf{Y}_n, \mathbf{Z}_n, \mathbf{Y}, \mathbf{Z})$ is a structural loss function to penalize a hypothesized parse tree $(\mathbf{Y}, \mathbf{Z})$ different from ground truth annotation $(\mathbf{Y}_n, \mathbf{Z}_n)$. Similar to [31], we adopt a relative loss as in Equ. (11), i.e., the loss of hypothesized parse tree relative to the best one $(\mathbf{Y}^*, \mathbf{Z}^*)$ that could be found from the candidate pool. That is

$$\Delta(\mathbf{Y}_n, \mathbf{Z}_n, \mathbf{Y}, \mathbf{Z}) = \delta(\mathbf{Y}_n, \mathbf{Z}_n, \mathbf{Y}, \mathbf{Z}) - \delta(\mathbf{Y}_n, \mathbf{Z}_n, \mathbf{Y}^*, \mathbf{Z}^*), \quad (11)$$

where $\delta(\mathbf{Y}, \mathbf{Z}, \mathbf{Y}', \mathbf{Z}') = \sum_{p \in \mathcal{T}} IoU(s_{y_p}^{p,z_p}, s_{y_p'}^{p,z_p'})$ is a function of measuring the part segmentation difference between any two parse trees $(\mathbf{Y}, \mathbf{Z})$ and $(\mathbf{Y}', \mathbf{Z}')$. In this paper, we employ a commonly-used cutting plane algorithm [11] to solve this structural max-margin optimization problem of Equ. (10).

5.2. Greedy Pruning Inference Algorithm

On the inference of AOG models, dynamic programming or inside-outside algorithm is commonly used in literature [22, 38]. However, the existence of side-way edges makes many loopy cliques in our AOG model, and thus prohibits the use of dynamic programming algorithm to efficiently infer the global optimum. In this paper, we adopt a modified dynamic programming with greedy pruning algorithm for model inference. This algorithm has a bottom-up scoring step followed by a top-down backtracking step.

In the first step, we recursively calculate the vertice's scores of our AOG in a bottom-up manner. For each part $p \in \mathcal{T}$, we compute the score on every combination of its candidate segment proposal and part type (y_p, z_p) according to Equ. (5), and retain a top-k list of the highest scored configuration values of (y_p, z_p). For each part composition $c \in \mathcal{N}$, we compute the scores of subgraph rooted by vertex c for candidate state configuration values based on Equ. (6), and also retain the top-k scored configuration values of $(y_c, z_c, \{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\})$. Particularly, only the top-k configuration values of (y_μ, z_μ) are considered in further calculation for each child vertex $\mu \in Ch(c, z_c)$. Thus the computational complexity for vertex c is proportional to the number of candidate state configuration values (i. e., $\sum_{z_c=1}^{K_c} k^{|Ch(c,z_c)|}$), where $|Ch(c, z_c)|$ denotes the number of child And-nodes linked to the Or-node for state z_c. In this paper, we set $k = 10$ and $|Ch(c, z_c)| \leq 3$, allowing the inference procedure tractable with a moderate quantity of state configuration values for each vertex. It does not harm the performance notably in practice, although this greedy pruning operation gets rid of a large fraction of candidate state

configuration values for part composition vertice (especially for the high-level ones, e.g. upper-body, lower-body). We validate this issue via a diagnostic experiment in Sec. 6.

After finishing the scoring of root vertex (i.e., whole human body) in the first step, we trigger another step to back-track the optimum state value from the retained top-k list for each vertex in a top-down manner. Concretely, for each part composition vertex c we select the best scored state configuration value of $(y_c, z_c, \{(y_\mu, z_\mu) : \mu \in Ch(c, z_c)\})$, and recursively infer the optimum state values of the selected child vertice given each $\mu \in Ch(c, z_c)$ as the root vertex of a subgraph. In the end, we can obtain the best parse tree from the pruned solution space of our AOG, and thus output corresponding state values of $(\mathbf{Y}, \mathbf{Z})$ to produce the final parsing result.

6. Experiments

In this section, we describe the details of our algorithm and conduct various experiments to demonstrate the effectiveness of our pose information in each stage of our pipeline.

6.1. Implementation detail

In part proposal selection (Sec. 3.2), we train linear SVR models for $P = 10$ part categories and select top $n_p = 10$ segments for each part category, as candidates of the final assembling stage. We treat left part and right part as two different part categories. Although the candidate segment pool could be type-dependent for a part/part composition in the AOG, we use common segment proposals in practice, which can significantly facilitate to compute the geometric compatibility features of side-way segment pairs.

For the segment feature in Sec. 3.3, we first normalize each kind of feature independently, then concatenate them together and normalize the whole feature. All the normalization is done with $L2$ norm. For simplicity, we only train one SVR model $g^p(s_i|\mathcal{L}, \mathcal{H})$ for each part category p in Sec. 3.2 so that $g_{z_p}^p = g^p, \forall z_p \neq 0$ in Equ. (5). However, due to the weight parameter $w_{z_p}^p$ is learned dependent on the part type z_p in training the AOG, the unary terms of different part types are type-specific in the AOG model.

6.2. Datasets and investigations

Data. We evaluate our algorithm on the Penn-Fudan benchmark [27], which consists of pedestrians in outdoor scenes with much pose variation. Because this dataset only provides testing data, following previous works [2, 20, 18], we train our parsing models using the HumanEva dataset [21], which contains 937 images with pixel-level label maps for parts annotated by [2]. The labels of two datasets are consistent, which include 7 body parts: { hair, face, upper-clothes, lower-clothes, arms (arm skin), legs (leg skin), and shoes }. For the pose model, we use the model provided by [5], trained on the Leeds Sports Pose Dataset [12].
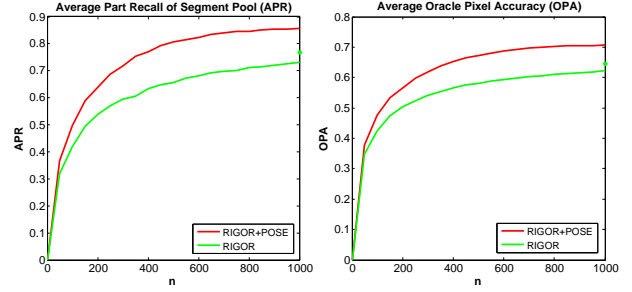

Figure 8. Comparison of our part segment proposal (Red curve) to RIGOR [10] (Green curve) on human parsing over two criteria w.r.t. the number of proposals. The green asterisks on the plots represent the APR/AOI of the RIGOR pool when the pool size $n = 2000$.

Effectiveness of pose for part proposal generation. We first investigate how our pose information help the initial part proposal generation. Specifically, we compare our pose-guided segment proposal method with the baseline proposal algorithm, i.e. the RIGOR algorithm [10] which is a faster substitute of the CPMC proposal [3] typically used by previous parsing approaches [7].

For evaluating the proposal algorithms, two standard criteria are used, i.e. average part recall (APR) and average part oracle IoU (AOI). The first measures how much portion of the ground truth segments is covered by the proposals, and the second measures the best IoU we can achieve given the proposals. Formally, the APR and AOI can be written as follows,

$$APR = \frac{1}{N} \sum_{i=1}^{N} \frac{|\mathcal{S}_i \cap \mathcal{G}_i|}{|\mathcal{G}|}, AOI = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{j \in \mathcal{G}_i} \max_{m \in \mathcal{S}_i} IoU(g_j, s_m)}{|\mathcal{G}_i|},$$

where N is the number of testing images. $\mathcal{S}_i$ and $\mathcal{G}_i$ are the set of segment proposals and the set of part segment ground truth in image i respectively. For computing $\mathcal{S}_i \cap \mathcal{G}_i$, we regard two segments as the same if their IoU is above 0.5.

In Fig. 8, we show the evaluated results over the Penn-Fudan test data. Specifically, we plot the APR and AOI w.r.t. the number of proposals upto 2000 segments. As shown in the figure, comparing with the RIGOR algorithm, ours (RIGOR + POSE) significantly improves the quality of part segment proposal by over 10% in average, which contributes much to our final performance. Finally, for each image, we select around 800 non-similar segments from the 2000 proposals as stated in Sec. 3.1. In Tab. 3, we list the APR and AOI score of the segment pool composed of the selected 800 segments, capable of achieving the accuracy as high as that of the original 2000 proposals.

Effectiveness of features for part proposal selection. To investigate various features proposed in Sec. 3.2 and their

complementary properties, we sequentially add features into our model and show the performance of the selected part segments. Specifically, the feature combinations we tested includes: (1) O2P + skin; (2) O2P + skin + PBG; (3) O2P + skin + PBG + C-PBG; (4) O2P + skin + PBG + C-PBG + deep potential, which we call Model 1 to Model 4 respectively.

The results are shown in Tab. 4, where we report the AOI score using the set of top-1 ranked part segment for each part class and top-10 ranked part segments. Firstly, we can see the results are sequentially improved, which demonstrate the effectiveness of all features we proposed. By comparing (2) and (3), we can see a significant boost of the top-1 accuracy, which indicates that after we coded the pose feature, the pose information becomes much more effective in the model. Finally, by adding the deep potential in (4), the performance of selected part segment is further improved.

We hence adopt (4) as our part ranking model to rank and select part proposals, yielding top n_p ranked candidates per part category for the final assembling. The quality of selected part proposals is evaluated in Tab. 5. We set $n_p = 10$ because it strikes a good balance between recall and segment pool size, and the oracle assembling result shows that we could surpass the state-of-the-art by over 15% by using only the selected part segment proposals.

	hair	face	u-cloth	l-cloth	arms	legs	shoes	mean
Recall	0.88	0.90	0.99	0.99	0.86	0.87	0.67	0.88
IoU	0.73	0.74	0.85	0.86	0.67	0.72	0.56	0.73

Table 3. Part recall and average oracle IoU of our segment pool.

Methods	hair	face	u-cloth	l-cloth	arms	legs	shoes	mean
o2p+skin	57.1	53.5	70.9	70.9	26.6	20.4	15.6	45.0
	68.8	66.9	80.0	81.4	54.6	55.3	45.3	64.6
(1)+PBG	61.7	58.6	73.2	72.7	29.9	23.4	17.5	48.1
	69.9	66.4	80.6	82.3	56.4	54.3	45.8	65.1
(2)+C-PBG	61.8	58.9	73.2	71.9	39.8	44.8	26.5	53.8
	69.9	66.4	80.5	82.4	55.8	59.1	47.4	65.9
(3)+deep potential	64.4	59.0	77.4	77.1	41.4	43.6	35.1	56.9
	70.7	66.6	82.2	83.4	55.9	59.3	48.8	66.7

Table 4. Comparison of four part models by AOI score (%) for top-1 ranked segment (top) and top-10 ranked segments (bottom). Models are numbered as (1) to (4), from top to bottom.

	hair	face	u-cloth	l-cloth	arms	legs	shoes	mean
Recall	0.86	0.79	0.99	0.99	0.68	0.69	0.55	0.79
IoU	0.71	0.67	0.82	0.83	0.56	0.59	0.49	0.67
Oracle	0.73	0.70	0.83	0.84	0.62	0.64	0.45	0.69

Table 5. Evaluation of our selected segment pool (top-10 segments per category) in terms of part recall, AOI score, and oracle assembling pixel accuracy.

Methods	hair	face	u-cloth	arms	l-cloth	legs	Avg
Naive Assembling	62.3	53.5	77.8	36.9	78.3	28.2	56.2
Basic AOG	63.1	52.9	77.1	38.0	78.1	35.9	57.5
Ours	**63.2**	**56.2**	**78.1**	**40.1**	**80.0**	45.5	**60.5**
Ours (w/o pruning)	63.2	56.2	78.1	40.1	80.0	**45.8**	60.5

Table 6. Per-pixel accuracy (%) of our AOG and two baselines.

Effectiveness of the AOG. To show the effectiveness of our AOG design, we set up two experimental baselines for comparison. (1) Naive assembling: consider only the unary terms and basic geometric constraints as defined in [2], e.g. upper-clothes and lower-clothes must be adjacent. (2) Basic AOG: consider only the unary terms and the vertical edges, without pairwise side-way edges from the C-PBG feature in Eqn. (5). We show the results in Tab. 6, where the Basic AOG with vertical relations outperforms simple hard constraints, and by adding the pairwise side-way edges, the results are further boosted, which strongly supports each component of our AOG model. We also display the results for our model without pruning for comparison, which clearly justifies our use of pruning in AOG inference. We can see that pruning only brings ignoreable decrease in performance while it reduces the inference time from 2 min. to 1 sec. per image.
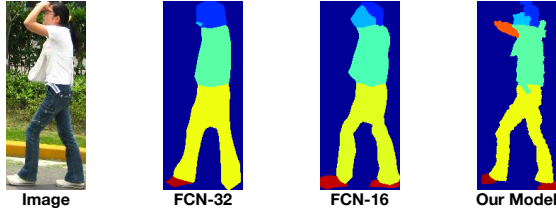
Training and testing time. Generally, for training, our approach takes two days for training a FCN model and two day for training the SVR selection model and AOG model. For testing, it takes 6s for extracting various features and around 1s to assemble the segments using AOG with our MATLAB implementation.

6.3. Comparison to the state-of-the-art

We compare our approach with four state-of-the-art methods, namely FCN [17], SBP [2], P&S [20], and DDN [18], which use the same training and testing settings. Specially, for FCN, we use the code provided by the author and re-train a model with our training set. Following [28], we train the FCN up to 16s version since there is little improvement using the 8s version. We refer the reader to [17, 28] for more details due to the limited space.

Method	hair	face	u-cloth	arms	l-cloth	legs	shoes	Avg*
FCN-32 [17]	50.2	33.7	69.4	13.8	66.7	14.2	25.2	41.3
FCN-16 [17]	48.7	49.1	70.2	33.9	69.6	29.9	**36.1**	50.2
P&S [20]	40.0	42.8	75.2	24.7	73.0	46.6	-	50.4
SBP [2]	44.9	**60.8**	74.8	26.2	71.2	42.0	-	53.3
DDN [18]	43.2	57.1	77.5	27.4	75.3	**52.3**	-	56.2
Ours	**63.2**	56.2	**78.1**	**40.1**	**80.0**	45.5	35.0	**60.5**

Table 7. Comparison of our approach with other state-of-the-art algorithms over the Penn-Fudan dataset. The Avg* means the average without shoes class since it was not included in other algorithms.

(a) Visual comparison between ours and FCN [28]

(b) Additional parsing results.

Figure 9. Qualitative results of our algorithm over the Penn-Fudan dataset [27].

Quantitative results. We show the comparison results in Tab. 7. Our model outperforms the potential directly from FCN by over 10% and the state-of-the-art DDN model by over 4%, from which we can see most improvement is from the small part such as hair and arms. It demonstrates that our pose information produces strong segment candidates that align to boundaries for small parts. In addition, our pose feature design together with deep potentials and AOG models obtains long-range context information, which gives our model robustness in shape variation and avoid many local confusions.

Figure 10. Failure cases of our algorithm on Penn-Fudan dataset. For each case, the original image (with pose prediction), ground truth, and our parsing result are displayed from left to right.

Qualitative results. As the authors of other comparisons did not release their code and visual results, qualitatively, we can only compare our results with the FCN as shown in Fig. 9(a). From the results, FCN can produce high accuracy over the general shape of the human while missing align to some local details like arms and shoes. Ours solve such problems with pose-guided local segment proposal, robust segment selection and assembling, yielding more satisfied results.

In Fig. 10, we show three failed examples due to color confusion with other objects, multiple instance occlusion, and big variation in lighting respectively, which generally fail most of the curent algorithms. For the first case and the third case, we got accurate pose prediction but failed to generate satisfying segment proposals for lower-clothes, which suggests that we either adopt stronger shape cues in the segment proposal stage or seek richer context information (e.g. handbag in the first case). For the second case, we got a bad pose prediction in the beginning and thus mixed two people's parts during assembling, which indicates the necessity of handling instance pose estimation or segmentation, which is beyond the scope of this paper.

7. Conclusion

In this paper, we present a human parsing pipeline which integrates human pose information and deep-learned features into segments, producing robust human parsing results. Our approach majorly includes three stages: part segment proposal, part proposal selection, and part assembling with an And-Or graph. In this framework, we systematically explore human pose information, including the pose-guided proposal and novel pose-based features which successfully applied to the part selection and assembling. We did extensive experiments that validate the effectiveness of each components of our framework, and finally, our approach significantly surpasses other state-of-the-art methods on the popular Penn-Fudan benchmark [27].

The future work includes several aspects: (1) Adopt useful shape cues for the part proposal and selection stages. (2) Use the learned part prototypes to define part sub-types and explicitly incorporate them into the AOG model; (3) Combine CNN with graphical models in a more efficient way to better utilize their complementary role in the human parsing task.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009. 2

[2] Y. Bo and C. C. Fowlkes. Shape-based pedestrian parsing. In *CVPR*, 2011. 1, 2, 3, 9, 10

[3] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 34(7):1312–1328, 2012. 3, 9

[4] J. a. Carreira, R. Caseiro, J. Batista, and C. Sminchis-escu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 3, 4, 7

[5] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014. 1, 3, 9

[6] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *BMVC*, 2011. 1

[7] J. Dong, Q. Chen, X. Shen, J. Yang, and S. Yan. Towards unified human parsing and pose estimation. In *CVPR*, 2014. 1, 2, 3, 9

[8] J. Dong, Q. Chen, W. Xia, Z. Huang, and S. Yan. A deformable mixture parsing model with parselets. In *ICCV*, 2013. 2

[9] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 3, 7

[10] A. Humayun, F. Li, and J. M. Rehg. RIGOR: Reusing Inference in Graph Cuts for generating Object Regions. In *Computer Vision and Pattern Recognition (CVPR), Proceedings of IEEE Conference on*. IEEE, june 2014. 2, 3, 9

[11] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009. 8

[12] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12. 9

[13] R. Khan, A. Hanbury, , and J. Stottinger. Skin detection: A random forest approach. In *ICIP*, 2010. 3

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR 2006*, pages 2169–2178, 2006. 7

[15] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *ICCV*, 2011. 7

[16] S. Liu, X. Liang, L. Liu, X. Shen, J. Yang, C. Xu, L. Lin, X. Cao, and S. Yan. Matching-cnn meets knn: Quasi-parametric human parsing. In *CVPR*, 2015. 1, 3

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, Nov. 2015. 2, 3, 4, 10

[18] P. Luo, X. Wang, and X. Tang. Pedestrian parsing via deep decompositional network. In *ICCV*, 2013. 1, 3, 9, 10

[19] L. Ma, X. Yang, Y. Xu, and J. Zhu. Human identification using body prior and generalized emd. In *ICIP*, 2011. 1

[20] I. Rauschert and R. T. Collins. A generative model for simultaneous estimation of human body shape and pixel-level segmentation. In *ECCV*, 2012. 9, 10

[21] L. Sigal and M. J. Black. Synchronized video and motion capture dataset for evaluation of articulated human motion. *Technical Report CS-06-08, Brown University*, 2006. 9

[22] J. M. Siskind, J. J. Sherman, I. Pollak, M. P. Harper, and C. A. Bouman. Spatial random tree grammars for modeling hierarchical structure in images with regions of arbitrary shape. *T-PAMI*, 29(9), Sept. 2007. 8

[23] S. Tsogkas, I. Kokkinos, G. Papandreou, and A. Vedaldi. Semantic part segmentation with deep learning. *arXiv preprint arXiv:1505.02438*, 2015. 3, 7

[24] D. A. Vaquero, R. S. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk. Attribute-based people search in surveillance environments. In *WACV*, 2009. 1

[25] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *CVPR*, 2013. 1

[26] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010. 7

[27] L. Wang, J. Shi, G. Song, and I. F. Shen. Object detection combining recognition and segmentation. In *ACCV*, 2007. 1, 2, 9, 11

[28] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Joint object and part segmentation using deep learned potentials. *arXiv preprint arXiv:1505.00276*, 2015. 3, 7, 10, 11

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

[30] M. Weber, M. Bauml, and R. Stiefelhagen. Part-based clothing segmentation for person retrieval. In *AVSS*, 2011. 1

[31] P. Yadollahpour, D. Batra, and G. Shakhnarovich. Discriminative re-ranking of diverse segmentations. In *CVPR*, 2013. 8

[32] K. Yamaguchi, M. H. K. Luis, E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012. 1, 2, 3

[33] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801. IEEE, 2009. 7

[34] M. Yang and K. Yu. Real-time clothing recognition in surveillance videos. In *ICIP*, 2011. 1

[35] W. Yang, P. Luo, and L. Lin. Clothing co-parsing by joint image segmentation and labeling. In *CVPR*, 2014. 1, 2, 3

[36] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 3

[37] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu. Action recognition with actons. In *ICCV*, 2013. 1

[38] J. Zhu, T. Wu, S.-C. Zhu, X. Yang, and W. Zhang. Learning reconfigurable scene representation by tangram model. In *WACV*, 2012. 2, 8

[39] L. Zhu, Y. Chen, Y. Lu, C. Lin, and A. Yuille. Max margin and/or graph learning for parsing the human body. In *CVPR*, 2008. 2

[40] S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):259–362, 2007. 2