OpenPose : Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields

Abstract

Realtime multi-person 2D pose estimation은 매우 중요! → image에서 다수의 사람들의 2D pose를 realtime으로 detect 해보겠다!

⇒ *nonparametric representation(Part Affinity Fields, PAFs)* 사용해서 했음 : 사진에서 개인의 body part 학습하기 위해

이런 bottom-up 방법 : 사진 속에 사람이 몇 명이 있던지, high accuracy 와 realtime performance 보여줌

- 이전 연구 : PAFs & body part location 예측이 training 에서 동시에 refine → PAFs만 refine 했더니 runtime performance & accuracy ↑
- body and foot keypoint detector 결합 : internal annotated foot dataset 사용 → detector 각각을 연속적으로 돌리는 것 보다 inference 감소 & 각 component의 accuracy 유지

Fig. 1: **Top:** Multi-person pose estimation. Body parts belonging to the same person are linked, including foot keypoints (big toes, small toes, and heels). **Bottom left:** Part Affinity Fields (PAFs) corresponding to the limb connecting right elbow and wrist. The color encodes orientation. **Bottom right:** A 2D vector in each pixel of every PAF encodes the position and orientation of the limbs.

Introduction

이미지와 비디오에서 detailed understanding 얻는 core component

: human 2D pose estimation 혹은 anatomical key point / part localizing

human estimation : 개개인들의 body part 찾는 것에 주로 집중

→ 사진에서 다수의 사람들의 pose 분석 매우 challengable

1. 각 사진에서 몇 명의 사람이 어떤 위치와 크기로 나타나는지 모름
2. interaction btw people (접촉한 경우, 가려진 경우, limb articulation) : complex spatial inference 야기 가능
   → 각 part들의 연관성 어렵게 만듦
3. runtime complexity가 사진 속의 사람수와 비례하는 경향 있음 → runtime performance 어렵게 만듦

- 일반적으로, person detector : 각 detection에 대해, single person pose estimation 수행

  → top-down 방식 : 현재 있는 single-person pose estimation 방식 사용
  - early commitment에 의해 문제 발생 : 사람이 가까이 있는 경우 detector fail → recovery 의지 X
  - runtime이 image에 있는 사람 수에 비례(detection한 사람 의미) ∵ single-person pose estimator 작동하기 때문

cf) top-down 방식

1. human detector 이용하여 사람 검출
2. 검출된 사람의 위치로부터 이미지 crop → pose estimation module에 넘김
3. **검출된 사람 각각에 대해 pose 찾도록 반복 수행**

- bottom-up : early commitment에 robust & runtime complexity가 image의 사람수와 비례 X
  - 현재까지는, 다른 body part 와 다른 사람들로부터 얻은 global contextual cue 직접적으로 사용 X
  - but initial bottom up : 이미지당 몇 분씩 소요되는 global inference 요구 → 비용 많이 들음 ⇒ 효율의 이점 사용 X

cf) bottom-up 방식

1. 모든 사람의 각 관절의 위치로 예상되는 부분들을 찾아냄
2. 각 관절의 위치를 이어 각 개별 인물들에 해당하는 관절 위치로 재생성

In this paper : multi-person pose estimation 에서 효율적인 방법 보여줌

- Part Affinity Fields(PAFs) 통해 association score 얻어 bottom-up representation 했음

  : image domain 에서 limb의 location 과 orientation을 encode하여 2D vector fields에 나타낸 것

detection 과 association을 동시에 bottom-up에 표현하는 것 : greedy parse에 대해 충분한 global context encoding 가능 → 낮은 computation cost로 높은 quality 결과 얻을 수 있음

> greedy algorithm 참고해서 생각할 것!
>
> greedy algorithm : 미리 정한 기준에 따라서 매번 가장 좋아 보이는 답을 선택하는 알고리즘

이것의 이전 버전 : "Realtime multi-person 2d pose estimation using part affinity fields " 논문 참고

→ 이 논문은 새로운 contribution 제시

1. PAF refinement : accuracy 올리는데 중요함 / body part prediction refinement 는 중요하지 않음

    - network depth 올리고, body part refinement stage 제거 (Section 3.1, 3.2 참고)

      → speed 2배, accuracy 7% 올림 (Sec 5.2, 5.3)
2. foot dataset with 15K human foot instance 사용(Section 4.2 참고) & body and foot keypoint 에 대한 model 결합 → body-only model의 속도와 정확도 유지 가능 (Sec5.5)
3. 위 방식의 generality 가능함 보여줌 : vehicle keypoint 예측 문제에 적용가능 (Section 5.6 참고)
4. OpenPose로 release!

Related Work

Single Person Pose Estimation

traditional approach : body part의 local observation 과 그것들 사이의 spatial dependency 사용하여 추론

spatial model : tree-structured graphical model 기반 → kinematic chain/non-tree model을 따라 인접한 부분 사이의 spatial 관계를 parametrically encoding → occlusion, symmetry, long range relationship 잡아내기 위해 추가적인 edge를 tree structure에 추가

body part에 대한 믿을 만한 local observation을 얻기 위해 CNN 많이 사용

Multi-Person Pose Estimation

multi person pose estimation을 위해 주로 top-down 방식 사용

1. 먼저 사람 detect
2. 각 detect 된 region에 대해 사람마다 독립적으로 pose estimation

→ 1명의 사람에 대해서는 적용가능 but early commitment 문제 & global inference가 필요한 다른 사람들 사이의 spatial dependency capture fail

> early commitment 무엇?

→ 몇몇 방식은 inter-person dependency 고려

"We are family: Joint pose estimation of multiple persons " : interacting 하는 사람과 순서를 고려하기 위해 extended pictorial structure 필요 but 여전히 초기 detection 위해 person detector 필요함

"Deepcut: Joint subset partition and labeling for multi person pose estimation" : bottom-up 방식 제안 → part detection candidate 와 각 사람과 candidates 간의 관련성을 labeling (detected part 간의 spatial offsets으로 부터 구해지는 pairwise score 사용)

이전 연구 : *part affinity fields(PAFs)* 제안 : 다수의 사람들의 body part 사이의 unstructured pairwise relationship encode한 set of flow fields

→ 이제는 추가적인 training step 없이 PAFs로 pairwise score 얻을 수 있음! ⇒ multi person estimation 위한 realtime performance에서 high quality 얻기 위해서는 충분

In this work, 이전 연구를 더 확장시켰음!

- PAF refinement가 high accuracy 위해 중요하고 충분하다는 것 보여줌 → network depth 증가시키면서 body part confidence map refinement 제거

  → 더 정확하고 빠른 model 만들어냄
- combined body and foot keypoint detector 보여줌

  → 두 detection 방법 결합 : 두 개를 각각 실행했을 때 보다 inference 시간 줄어듦 & 각각의 accuracy 유지
- OpenPose 보여줌 : real time body, foot, hand, facial keypoint detection 위한 open source library

Method

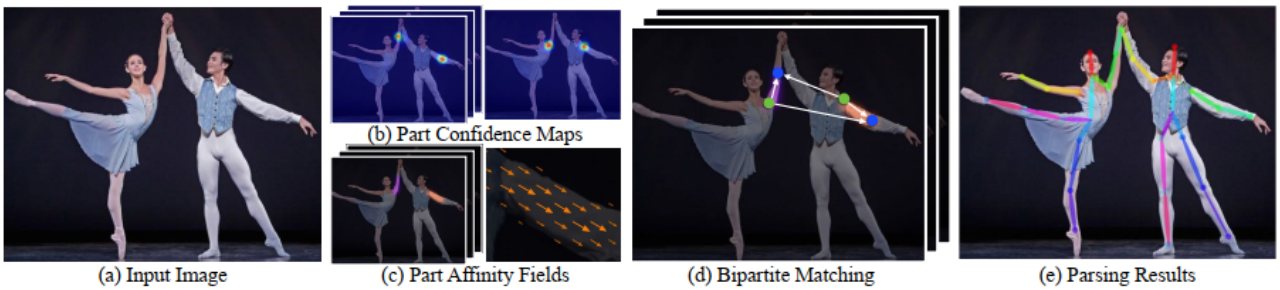


Fig. 2: Overall pipeline. (a) Our method takes the entire image as the input for a CNN to jointly predict (b) confidence maps for body part detection and (c) PAFs for part association. (d) The parsing step performs a set of bipartite matchings to associate body part candidates. (e) We finally assemble them into full body poses for all people in the image.

위의 그림이 논문 방법의 전체적인 pipeline을 보여줌

input : size $w \times h$ color image

output : 사진의 각 사람의 anatomical keypoint의 2D 위치정보

1. feedforward network
   - body part location의 2D confidence map $\mathbf{S}$

     $\mathbf{S} = (\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_J)$: J 개의 confidence map을 갖음 (각 part 마다 하나씩)

     $\mathbf{S}_j \in \mathbb{R}^{w \times h}, j \in \{1, \ldots, J\}$

     > 실제 코드에 보면 각 stage 별로 loss를 모두 구하고 이를 다 더해서 최종적인 branch의 loss를 구함!
     >
     > → 각 stage 에서 원하는 부분에 대한 confidence map을 구한다고 볼 수 있음

   - part affinity fiedls(PAFs)의 2D vector fields $\mathbf{L}$: parts 사이의 association 정도를 encoding

     $\mathbf{L} = (\mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_C)$: C개의 vector fields 갖음 (각 limb 마다 1개씩)

     $\mathbf{L}_c \in \mathcal{R}^{w \times h \times 2}, c \in \{1, \ldots, C\}$

   limb 같은 part pair를 분명성을 위해 언급 but 몇 pair들은 limb와 같지 않음 (ex. face)

   $\mathbf{L}_c$ 에 담긴 각 image location은 2D vector로 encoding!

   > 각 픽셀에 대해 벡터값을 구하는 것(x, y) → channel = 2
2. confidence map 과 PAFs : greedy inference에 의해 분석되어 이미지의 모든 사람들의 2D keypoint 제공

3.1 Network Architecture

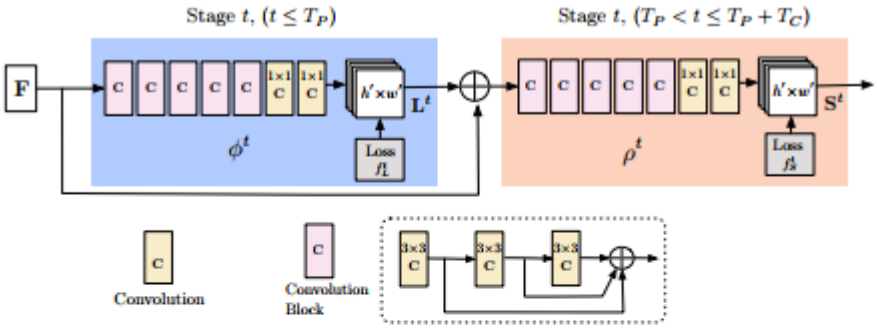


Fig. 3: Architecture of the multi-stage CNN. The first set of stages predicts PAFs $\mathbf{L}^t$, while the last set predicts confidence maps $\mathbf{S}^t$. The predictions of each stage and their corresponding image features are concatenated for each subsequent stage. Convolutions of kernel size 7 from the original approach [3] are replaced with 3 layers of convolutions of kernel 3 which are concatenated at their end.

- blue : 반복적으로 affinity fields 예측 → part-to-part association encode

  beige : detection confidence map 예측

  iterative prediction architecture : 연속적인 stage$t \in \{1, \ldots, T\}$를 통해 prediction 정밀화 (각 stage에서 intermediate supervision)

  > intermediate supervision 어떻게 하는 것?

- network depth 증가

  original approach : 여러개의 7×7 convolutional layers로 구성 ⇒ #operation = $2 \times 7^2 - 1 = 97$

  현재 model : 각 7×7 convolutional kernel을 3개의 연속적인 3×3 kernel로 대체 → computation 줄이고 receptive field 유지 ⇒ 51

- 각 3개의 convolutional kernels의 output을 concatenate → DenseNet과 비슷한 방식 사용

  ⇒ non-linearity layer 3배 & lower level 과 high level feature 모두 유지 가능

  ⇒ accuracy와 runtime speed 향상

3.2 Simultaneous Detection and Association

image : VGG-19의 초기 10개 layer 사용하여 분석 → first stage에 input으로 들어갈 set of feature maps $\mathbf{F}$ 생성

1. PAF 예측

   network가 a set of part affinity fields(PAFs) 생성

$$\mathbf{L}^1 = \phi^t(\mathbf{F}, \mathbf{L}^{t-1}), \forall 2 \leq t \leq T_P$$

ϕ^t : stage t에서 사용하는 inference를 위한 CNNs 의미 (CNN을 그냥 일종의 함수로 표현)

T_P : 전체 PAF stage 갯수

> T_P : 구하고자 하는 포인트? 의 갯수 x 2

T_P iteration 반복 후 (confidence map detection을 위해) → 가장 update 된 PAF 사용해서 confidence map detection

2. confidence map 예측

$$\mathbf{S}^{T_P} = \rho^t(\mathbf{F}, \mathbf{L}^{T_P}), \forall t = T_P$$
$$\mathbf{S}^t = \rho^t(\mathbf{F}, \mathbf{L}^{T_P}, \mathbf{S}^{t-1}), \forall T_P < t \leq T_P + T_C$$

ρ^t : stage t에서 사용하는 inference 위한 CNNs 의미

T_C : 전체 confidence map stage 갯수

- 각 stage에서 동시에 PAF와 confidence map 예측하던 것과는 다름

  → 각 stage에서 parameter 수 절반으로 줄음

- *경험적으로* affinity field prediction refine 한 것이 confidence map 결과 향상 ← 역으로는 아님

  직관적으로, PAF channel output 보면 body part location 예측 가능

  but bunch of body parts 가 다른 정보 없이 주어진다면, 각각을 다른 사람에게 맞춰 분석 불가능

Fig. 4: PAFs of right forearm across stages. Although there is confusion between left and right body parts and limbs in early stages, the estimates are increasingly refined through global inference in later stages.

위 그림이 각 stage 마다 affinity fields가 정밀화 되는 과정을 보여줌 → confidence map이 가장 최신으로 정밀화된 PAF prediction을 사용하여 예측됨!

- network가 첫번째 branch에서는 PAF를,두 번째 branch 에서는 confidence map을 predict 하기 위해 → 각 stage의 끝단에 loss function 적용

  : L2 loss 사용 (estimated prediction 과 map과 field의 groudtruth 사이의)

  → loss function spatially weight 줌! : 어떤 dataset은 모든 사람에 대해 완벽하게 labeling 되어 있지 않기 때문!

  - stage t_i에서 PAF branch의 loss function

$$f_{\mathbf{L}}^{t_i} = \sum_{c=1}^{C} \sum_{\mathbf{p}} \mathbf{W}(\mathbf{P}) \cdot \| \mathbf{L}_c^{t_i}(\mathbf{p}) - \mathbf{L}_c^*(\mathbf{p}) \|_2^2$$

  - stage t_k에서 confidence map branch의 loss function

$$f_{\mathbf{S}}^{t_k} = \sum_{j=1}^{J} \sum_{\mathbf{p}} \mathbf{W}(\mathbf{P}) \cdot \| \mathbf{S}_j^{t_k}(\mathbf{p}) - \mathbf{S}_j^*(\mathbf{p}) \|_2^2$$

  $\mathbf{L}_c^*$: groundtruth PAF , $\mathbf{S}_j^*$: groundtruth part confidence map

  $\mathbf{W}$: binary mask with $\mathbf{W}(\mathbf{p}) = 0$ pixel $\mathbf{p}$에 annotation이 없는 경우

- mask : training 중에 true positive prediction에 penalize하는 경우 막기 위해 사용
- 각 stage 마다 intermediate supervision : gradient를 주기적으로 보충해줌으로써 vanishing gradient 문제 해결
- **overall objective**

$$f = \sum_{t=1}^{T_P} f_{\mathbf{L}}^t + \sum_{t=T_P+1}^{T_P+T_C} f_{\mathbf{S}}^t$$

3.3 Confidence Maps for Part Detection

training 중에 $f_{\mathbf{S}}$ 평가 : annotated 2D keypoints를 통해 groundtruth confidence map인 $\mathbf{S}^*$ 생성

각 confidence map : 주어진 pixel에서 각각의 body part가 있을꺼라고 여겨지는 위치를 2D representation한 것

ex. 만약 이상적인 경우, 사진에 사람이 1명 있다면 : 각 confidence map의 peak → 실제 눈으로 봤을 때 대응하는 위치에 나타남

사진에 사람이 여러명 있다면 : 각 사람 k에 대해 각각의 j개의 part가 대응되어 peak로 나타남

- 사람 k에 대해 개별적인 confidence map $\mathbf{S}_{j,k}^*$ 생성

  $\mathbf{x}_{j,k} \in \mathbb{R}^2$: 사진의 k사람에 대한 body part j의 groundtruth 위치 의미

  $\mathbf{S}_{j,k}^*$에서의 $\mathbf{p} \in \mathbb{R}^2$ location의 value :

$$\mathbf{S}_{j,k}^*(\mathbf{p}) = exp(-\frac{\|\mathbf{p} - \mathbf{x}_{j,k}\|_2^2}{\sigma^2})$$

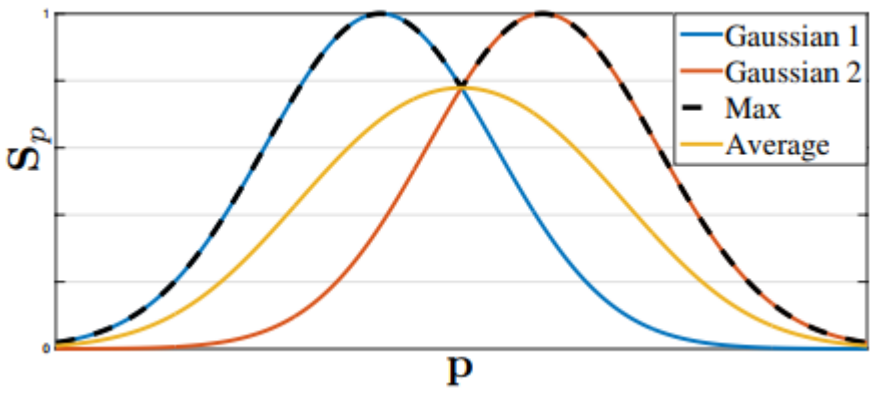
  σ : spread의 peak 다룸

  (peak location을 중심으로 가우시안 noise 넣은 것이라고 생각해도 될 것 같음 / 그래서 peak 값에서 값이 크고 주변으로 점점 가우시안으로 작아지는! 시그마의 경우 그 작아지는 정도를 얼마가 가파르게 할 것인가를 나타내주는 정도라고 이해하면 될 듯)

- network를 통해 예측한 confidence map의 groudtruth : 각 confidence map에 max 값만 취한 것의 집합으로!

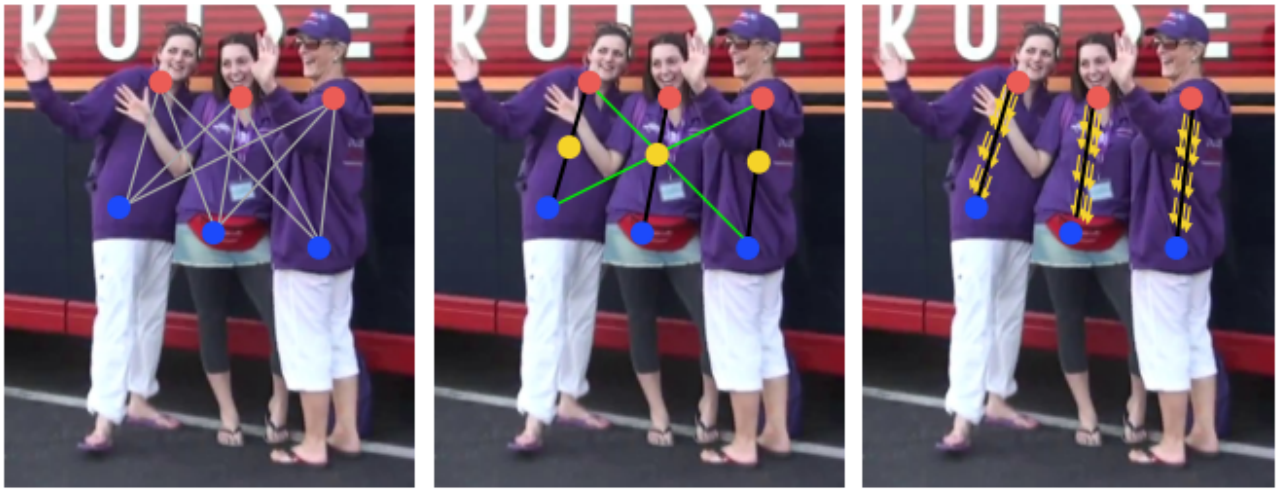
$$\mathbf{S}_j^*(\mathbf{p}) = \max_f \mathbf{S}_{j,k}^*(\mathbf{p})$$

  → average 대신 max 값 사용 : 가까운 peak의 precision 가깝게 유지 ⇒ 가까운 peak 구분 가능



- test시 : confidence map 예측 → body part candidates 얻음 (non-maximum suppression 사용)

3.4 Part Affinity Fields for Part Association



(a) (b) (c)

Fig. 5: Part association strategies. (a) The body part detection candidates (red and blue dots) for two body part types and all connection candidates (grey lines). (b) The connection results using the midpoint (yellow dots) representation: correct connections (black lines) and incorrect connections (green lines) that also satisfy the incidence constraint. (c) The results using PAFs (yellow arrows). By encoding position and orientation over the support of the limb, PAFs eliminate false associations.

set of detected body part 주어짐 (위 그림 a) → 몇명인지 모르는 사람들에게 각자에 맞게 맞춰줘야함!

⇒ body part detection의 각 쌍에 대해 관련성에 대해 confidence measure로 필요(각각이 같은 사람에 대한 것이라는 것을 나타내줘야함)!

- association measure 방법

  limb part의 쌍에 사이의 추가적인 midpoint detect & part detection 후보 사이의 incidence 확인(Fig 5-b)

  → 사람이 많은 경우 midpoint가 잘못된 association 줄 수 있음(b의 green line 같이)

  → 이런 잘못된 정보 : representation에서 2개의 limitation에 의해 발생

  ```
  1. 각 limb에 대해 position에 대해서만 encode되어 있고, orientation에 대해서는 x
  ```

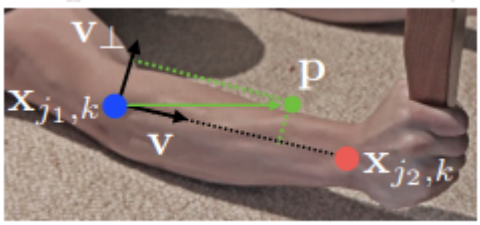
  2. limb가 support 해줄 수 있는 영역을 한 점으로 줄여버림

⇒ Part Affinity Fields (PAFs)가 위의 limitation들 해결!

: limb이 support 하는 영역에 걸쳐 location과 orientation information 동시에 보존!(Fig 5-c 참고)

- 각 PAF : 각 limb에 대한 2D vector field
- 각각의 limb에 속해있는 영역의 각 pixel : limb의 한 부분의 point 에서 다른 부분의 point로의 방향을 2D vector로 encode
- 각 limb의 type : 대응되는 PAF 갖음(관련된 2개의 body part 연결)

아래와 같이 사진 안에 1개의 limb만 있는 경우



$\mathbf{x}_{j_1,k}$ 와 $\mathbf{x}_{j_2,k}$: image의 사람 k에 대한 c limb의 body part j_1과 j_2의 groundtruth position

point $\mathbf{p}$가 limb 위에 있는 경우

: $\mathbf{L}^*_{c,k}(\mathbf{p})$의 value = point j_1에서 j_2로의 unit vector

: 그 외의 다른 모든 점에 대해서는 vector 값 0

- training 중 $f_\mathbf{L}$ 평가 위해 : groundtruth PAF 정의!
  - point $\mathbf{p}$에서 $\mathbf{L}^*_{c,k}$

$$\mathbf{L}^*_{c,k}(\mathbf{p}) = \begin{cases} \mathbf{v} & \text{if } \mathbf{p} \text{ on limb } c,k \\ 0 & \text{otherwise} \end{cases}$$

  $\mathbf{v} = (\mathbf{x}_{j_2,k} - \mathbf{x}_{j_1,k}) / \parallel \mathbf{x}_{j_2,k} - \mathbf{x}_{j_1,k} \parallel_2$: limb 방향의 unit vector

  - limb의 point들의 set : line segment의 distance threshold 사용

    : those points $\mathbf{p}$ for which

$$0 \leq \mathbf{v} \cdot (\mathbf{p} - \mathbf{x}_{j_1,k}) \leq l_{c,k} \quad and \quad |\mathbf{v}_\perp \cdot (\mathbf{p} - \mathbf{x}_{j_1,k})| \leq \sigma_l$$

  limb width σ_l : pixel에서의 distance

  limb length $l_{c,k} = \parallel \mathbf{x}_{j_2,k} - \mathbf{x}_{j_1,k} \parallel$

  $\mathbf{v}_\perp$: vector perpendicular to $\mathbf{v}$

> limb에 해당하는 point들 : line seg에 threshold 거는 것으로 구함!
>
> $l_{c,k}$: x_2까지의 거리 내에 있어야함 (가로로!)
>
> σ_l : 팔에서 벗어나지 않는 범위 안에 있어야함 (세로로!)

- groundtruth part affinity field : 사진의 모든 사람들에 대한 affinity fields 의 평균

$$\mathbf{L}_c^*(\mathbf{p}) = \frac{1}{n_c(\mathbf{p})} \sum_k \mathbf{L}_{c,k}^*(\mathbf{p})$$

$n_c(\mathbf{p})$: 모든 k명의 사람들에 있는 point $\mathbf{p}$에서 non-zero vector들의 수

- testing시 : part detection candidates 사이의 association 측정 → candidate part location 연결하는 line segment 따라 대응되는 PAF 들 선적분

  : detected body parts 연결 → candidate limb 와 predicted PAF 사이의 alignment 측정

  part location d_{j_1} 와 d_{j_2} 후보 있음 → predicted part affinity field $\mathbf{L}_c$ (후보들의 association 의 confidence 측정하기 위해 line segment 수행)

$$(11) \quad E = \int_{u=0}^{u=1} \mathbf{L}_c(\mathbf{p}(u)) \cdot \frac{d_{j_2} - d_{j_1}}{\| d_{j_2} - d_{j_1} \|_2} du$$

$\mathbf{p}(u)$: d_{j_1} 과 d_{j_2} 사이 interpolate → $\mathbf{p}(u) = (1-u)d_{j_1} + ud_{j_2}$

> ??????????????
>
> E : part detetion candidate 사이의 association 의미

3.5 Multi-Person Parsing using PAFs

detection confidence maps에 non-maximum suppression 적용 : part candidate locations을 discrete 하게 얻기 위해!

각 part에 대해, several candidate 갖음 ← false negative 나 이미지에 여러명의 사람 있는 경우

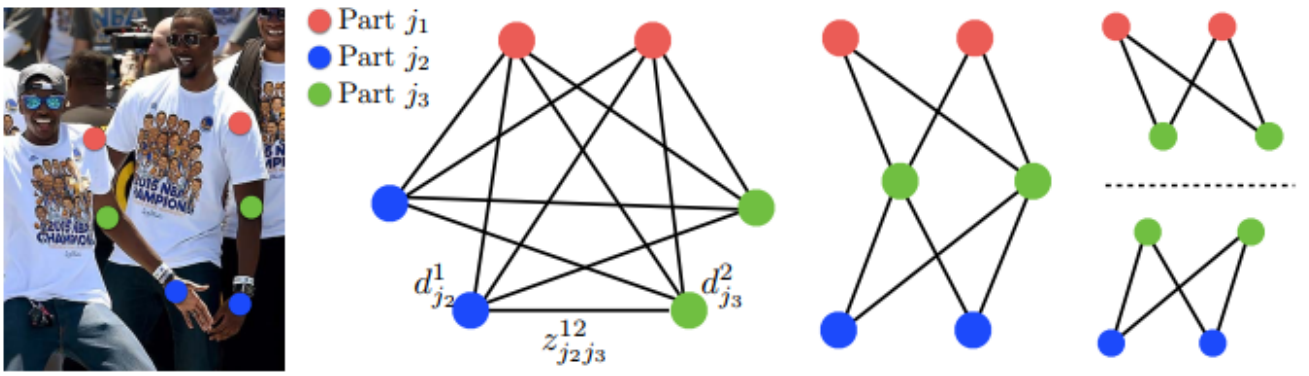


Fig. 6: Graph matching. (a) Original image with part detections. (b) K-partite graph. (c) Tree structure. (d) A set of bipartite graphs.

→ part candidates : 가능한 limbs의 큰 set 정의! ⇒ 각 candidate limb score! (11번 식 사용) ← PAF에서 선적분하여!

> part candidate 끼리의 association 확인함!

⇒ optimal parse 찾는 문제 : K-dimensional matching 문제와 대응 → NP-Hard!

- 계속적으로 high quality match 위해 : *greedy relaxation* 적용!

  문제 분석 해보았음!

  - pair-wise association score : global context encoding! (PAF network의 큰 receptive field 때문)
- 여러명의 사람에 대해 body part detection candidates $\mathcal{D}_{\mathcal{J}}$

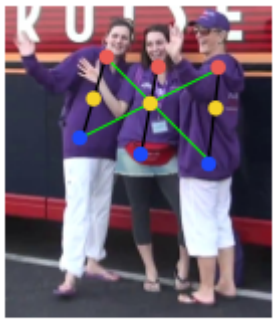
  $\mathcal{D}_{\mathcal{J}} = \left\{ d_j^m : for j \in 1...J, m \in 1...N_j \right\}$, N_j : part j의 후보들의 갯수, $d_j^m \in \mathbb{R}^2$: body part j에 대한 m번째 detection 후보의 위치

  → 같은 사람의 다른 part 와 관련되어야함! ⇒ 실제로 limbs와 연결된 detection part를 찾아야함

  > 총 N_j개의 part 존재 & 각 part 는 J개의 후보를 갖고 있음
- $z_{j_1 j_2}^{mn} \in \{0, 1\}$: 두 detection 후보 $d_{j_1}^m$과 $d_{j_2}^n$ 이 서로 연결되었는지를 나타내는 variable

  → Goal : 가능한 모든 연결들($\mathcal{Z} = \left\{ z_{j_1 j_2}^{mn} : \text{for} j_1, j_2 \in \{1...J\}, m \in \left\{1...N_{j_1}\right\}, n \in \left\{1...N_{j_2}\right\} \right\}$)에 대해 optimal assignment 찾는 것!
- 1개의 pair 만 있는 경우 고려 ex. 목 / 오른쪽 엉덩이

  : 적절한 association 찾는 문제 → bipartite graph 문제에서 weight maximum을 감소시키게 됨

  (b)

  옆의 그림과 같은 경우를 말하게 됨!

  graph의 노드 : body part detection candidates $\mathcal{D}_{j_1}$ 과 $\mathcal{D}_{j_2}$

  graph의 엣지 : detection candidates 사이의 모든 가능한 connection들

  → 각각의 edeg는 식 (11)에 의해 weighted 됨! (the part affininty aggregate)
- bipartite graph의 matching : 두 edge 사이에 노드가 겹지치 않는 경우에 대한 edge들의 부분집합!
- Goal : 선택한 edge들에 대해 maximum weight로 matching하는 것을 찾는 것!

$$(13) \quad \mathbf{max}_{\mathcal{Z}_c} E_c = \mathbf{max}_{\mathcal{Z}_c} \sum_{m \in \mathcal{D}_{j_1}} \sum_{n \in \mathcal{D}_{j_2}} E_{mn} \cdot z_{j_1 j_2}^{mn},$$

$$(14), (15) \quad s.t \quad \forall m \in \mathcal{D}_{j_1}, \sum_{n \in \mathcal{D}_{j_1}} z_{j_1 j_2}^{mn} \leq 1, \quad \forall n \in \mathcal{D}_{j_2}, \sum_{m \in \mathcal{D}_{j_1}} z_{j_1 j_2}^{mn} \leq 1$$

  E_c : limb c에 matching 된 모든 weight $\mathcal{Z}_c$: limb c에 대한 subset $\mathcal{Z}$

  E_{mn} : 식 11에 의해 정의된 part $d_{j_1}^m$ 과 $d_{j_2}^n$ 사이의 part affinity

  식 (14), (15) : node를 공유하는 두 개의 edge가 없도록 해줌 → 같은 종류의 part를 두 개의 limb이 공유하지 않도록 해줌!

  ⇒ optimal matching을 찾기 위해 *Hungarian algorithm* 사용

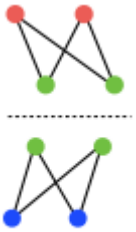
다수의 사람 사이에서 full body pose를 찾는 경우 → $\mathcal{Z}$ 결정하는 것 = *K-dimensional matching problem* ⇒ NP-hard!

⇒ 해결 위해 2개의 relaxation을 optimization에 추가해줌!

1. 전체 graph를 쓰는 대신, human pose의 spanning tree skeleton 얻기 위해 minimal number of edge만 선택!!

(c)

2. matching problem을 bipartite matching subproblems의 집합으로 분해 → 가까운 tree node들 각각에서 matching 결정!



(d)

⇒ Section 5.1에서 자세한 비교 결과 정리! : computational cost를 쪼갰을 때, minimal greedy inference 가 global solution 찾는 데 적절함을 보임!

∵ 가까운 tree nodes 사이의 relationship이 PAFs에 의해 명확하게 모델링 but 내부적으로 - 가깝지 않은 tree nodes 사이의 relationship은 CNN에 의해 대략적으로 모델링

→ CNN : 큰 receptive field로 학습 & 가깝지 않은 tree node 사이의 PAF : PAF 예측하는 데 영향을 줌!

위의 두 relaxation을 더하면 optimization이 아래와 같이 간단히 정리!

$$\mathbf{max}_{\mathcal{Z}} E = \sum_{c=1}^{C} \mathbf{max}_{\mathcal{Z}_c} E_c$$

식 (13) ~ (15) 를 사용하여 각 limb type에 대한 limb connection candidate 구함!

모든 limb connection candidates를 사용하여 : 다수의 사람들 사이의 full body pose 로 같은 part detection candidate를 공유하는 connection들을 조립!

⇒ tree structure에 대한 최적화 : fully connected graph에 대한 최적화 보다 더 빠름!

현재 모델 : 충분한 PAF connection들을 포함하고 있음 → 아래와 같이 사람이 많은 이미지에서 정확도를 높여주게 됨!

(a) Original Person Parsing (b) PAF-Redundant Parsing

Fig. 7: Importance of redundant PAF connections. (a) Two different people are wrongly merged due to a wrong neck-nose connection. (b) The higher confidence of the right ear-shoulder connection avoids the wrong nose-neck link.

→ 충분한 connection을 다루기 위해 : multi-person parsing algorithm 약간 수정!

: original - root component에서 시작! but 모든 가능한 pairwise connection들을 그들의 PAF score를 사용하여 분류

ex. connection이 이미 다른 사람에게 할당된 part 를 연결하려 한다면 - algorithm이 더 높은 confidence를 갖는 것을 선택! → 그 이전 것은 무시함