OP 特点(官网所列)

- **功能**:
  - **二维多人关键点实时识别**:
    - **15、18 或 25 个身体/脚部的关键点识别，运算时间与检测出的人数无关。**
    - *221 个手部关键点识别。目前，运算时间取决于检测出的人数*。*
    - **70 个面部关键点的识别。目前，运算时间取决于检测出的人数**
  - **三维单关键点实时识别**:
    - 通过多个单一角度的视频进行三角测量。
    - Flir 摄像机的视频同步处理。
    - 与 Flir 摄像机和 Point Grey 摄像机兼容，提供了 C++语言的代码样本，用户可以自定义输入。
  - **校准工具**:
    - 能够对摄像机拍摄中出现的扭曲等内外参数进行简易评估。
  - 针对未来的加速优化和视觉流畅，增加了**单人位置追踪** 。
- **输入**: 图片、视频、网络摄像头的视频流、Flir 或 Point Grey 和 IP 摄像机。项目提供了 C++语言的代码样本，用户可以自定义输入。
- **输出**: 原有图片+关键点展示（PNG、JPG、AVI 等格式），关键点数据存储文件（(JSON, XML, YML 等格式）。
- **操作系统**: Ubuntu (14, 16), Windows (8, 10), Mac OSX, Nvidia TX2.
- **其它**:
  - 项目提供: 命令行测试、C++封装、C++ API 接口。
  - CUDA (Nvidia GPU), OpenCL (AMD GPU), and CPU 版本。

OpenPose Paper 介绍

标题：

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

Abstract：

- Realtime approach (实时方法)
- Detect the 2D pose of multiple people in an image（可以检测一张图片中多人的 2D 姿态）
- Use a nonparametric representation (Part Affinity Fields (PAFs)),

to learn to associate body parts with individuals in the image.（使用了非参数表示方法（我不太了解）PAFs 即部分亲和字段，用来表示肢体间的联系）

- Body and foot key point detector and model（独特的身体与脚步关键点检测模型）
- Better runtime performance and accuracy（更好的性能/准确率）

现有的部分姿态识别挑战：

- Unknown number of people at any position（每张图片或每一帧中不同位置或某一范围内会有不确定的人数，随机人数会对检测要求更高）
- Interactions between people -> complex spatial interference. （人与人之间的交互会产生复杂的空间干扰，肢体的交叉或者缠绕给检测带来了困难）
- More number of people -> Runtime complexity （人数增多时很多算法的性能也会受到影响）

现有的两大类识别方法的特点/问题：

1.Top-down（自顶向下）

- People detector limitation
- Relation between runtime and number of people

自顶向下是首先识别出图片中的所有人，然后再识别出每个人对应的肢体，这种方法首先受限于一些 people detector 方法，假如识别人时就有问题那识别肢体时问题会更严重，因此这类方法要求高质量的人识别方法；其次由于需要找出图片中所有的人，因此人数增多时该类方法就会比较慢，效果可能受到影响。

2.Bottom-up（自底向上）

- Better robustness and time complexity
- Don't directly use global contextual cues from other body parts
- Global inference and worse efficiency

自底向上则是不需要识别图片中的所有人，先把所有肢体识别出来再进行拼接或联系，鲁棒性和效果也比较不错，但是缺点如同上面后两点所示。

Related Work：

- Single Person Pose：CNN eg.
- Multi Person Pose

相关研究中很多利用 CNN 进行姿态识别的方法效果不错但是仅适用于单人姿态识别，多人姿态识别如同上面两类方法所说存在一些问题。

基本模型（这一部分我也是半知半解）：

1. 文章使用了两种模型或者说定义，首先图片宽高定义为 w、h。
首先姿态识别首先需要识别出肢体的定位点，比如人体的手踝和肘部就各有一个定位点，这两个定位点中间的部分就是手臂前肢。即找到了定位点就相当于找到了相关的肢体。

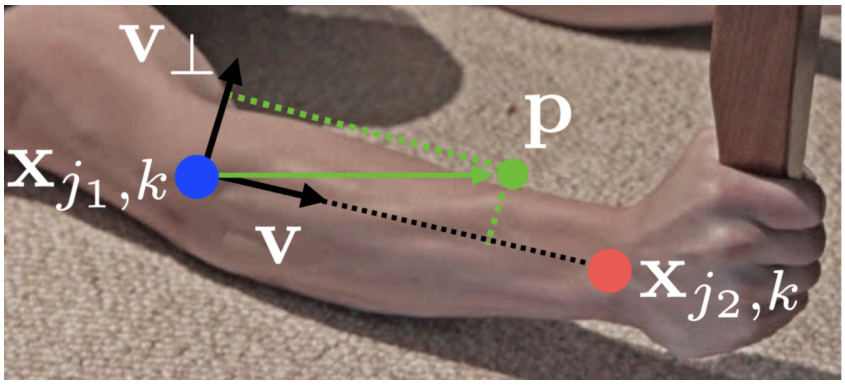
2. 关于 Part Confidence Map（部分置信映射，后用 PCM 简称）：
一张图片中第 k 个人的第 j 个肢体的定位点真实位置用 $x_{j,k}$ 表示，有一座标点 p，则该处的 PCM 由下定义，*号代表计算的基准是真实值（无*号的公式代表计算的基准是网络中预测的定位点位置）。显然 p 与 $x_{j,k}$ 差距越小，这个计算的 PCM 值越大。

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

由于本文是一个自底向上的方法，对计算的值进行合并（取最大值），代表 p 处是肢体定位点 j 的衡量值

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

3. 关于 Part Affinity Fields（部分亲和字段，后用 PAF 简称）：



我们用 $x_{j1,k}$ 和 $x_{j2,k}$ 来表示图中第 k 个人一个肢体 c 上两个定位点 j_1 与 j_2 点真实值，我们使用 v 表示单位向量，则定义 PAF：

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

P 是一个座标点，只有 p 在 k-th 人的肢体 c 上时 L 才是单位向量，否则为零向量。判断的条件如下（大致就是 p 点在这个肢体形成的矩形范围内）：

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

按照这种定义的结果，人肢体上存在很多这种单位向量，如下所示

$n_{c(p)}$ 代表 p 处非零向量的个数，如同 PCM 最后的归并，可以计算得

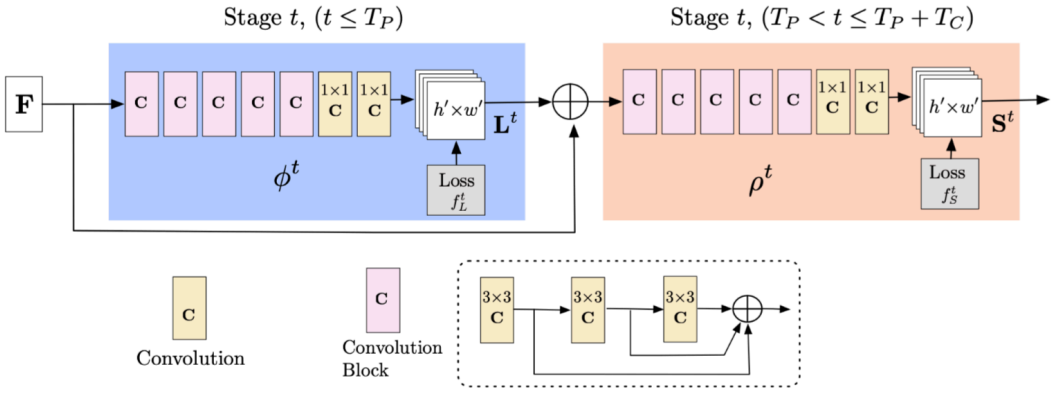
$$\mathbf{L}_c^*(\mathbf{p}) = \frac{1}{n_c(\mathbf{p})} \sum_k \mathbf{L}_{c,k}^*(\mathbf{p}),$$

作者使用下面的公式计算 d_j1、dj2 的联系值，dj1、dj2 为两个 candidate part locations （候选的肢体定位？），这部分计算借助了 PAFs

$$E = \int_{u=0}^{u=1} \mathbf{L}_c\left(\mathbf{p}(u)\right) \cdot \frac{\mathbf{d}_{j_2} - \mathbf{d}_{j_1}}{\left\|\mathbf{d}_{j_2} - \mathbf{d}_{j_1}\right\|_2} du,$$

$$\mathbf{p}(u) = (1-u)\mathbf{d}_{j_1} + u\mathbf{d}_{j_2}.$$

结构：



搭建的网络结构如上，L^t 代表 PAFs，S^t 代表 PCMs。首先借助下面论文的网络的前十层从图片中提取特征集 F：

K. Simonyan and A. Zisserman, “Very deep convolutional net- works for

large-scale image recognition," in ICLR, 2015.

在蓝色部分，由 F 得到 PAFs 后，再将其与特征重新作为输入重复蓝色部分，反复得到更新的 PAFs，下面的 t 代表阶段 stage t

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

米色部分与前面类似，首先以 F 与 PAFs 作为输入经网络得到 PCMs，再将结果作为新的输入不断得到更新的 PCMs：

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

为什么采取这种迭代的方式以及先生成 PAF 后生成 PCM 文章后面用数据证明这种方法效果比较好。
这一部分文中并未详细介绍过程，只是讲解了大致结构。采用 L_2 Loss 作为损失函数：

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

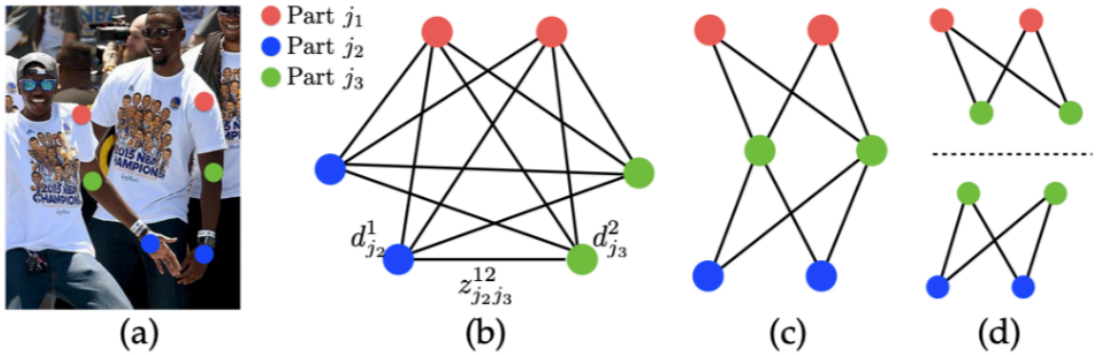
$$f_{\mathbf{S}}^{t_k} = \sum_{j=1}^{J} \sum_{\mathbf{p}} \mathbf{W}(\mathbf{p}) \cdot \|\mathbf{S}_j^{t_k}(\mathbf{p}) - \mathbf{S}_j^*(\mathbf{p})\|_2^2,$$

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

连接：

(a) (b) (c)

如上图所示，如何将图中各点像图 c 一样正确连接成一个人的肢体（图中的肢体是躯干）需要特殊的方法。

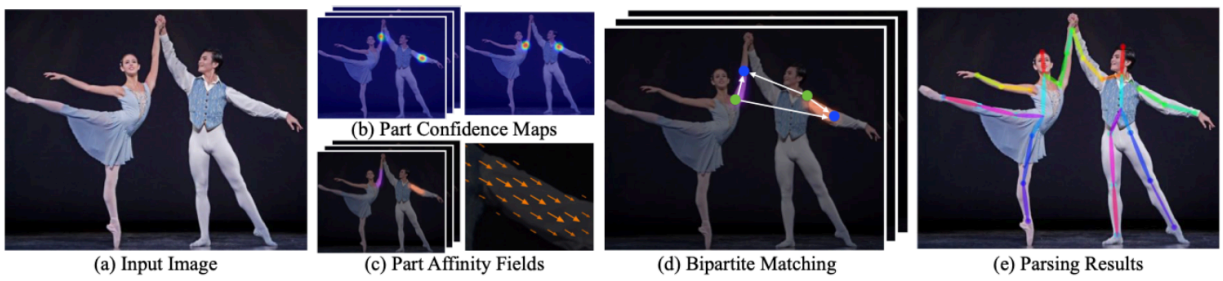


文中给出的方法大致总结为

1. Choose a minimal number of edges to obtain a spanning tree skeleton of human pose rather than using the complete graph
2. Decompose the matching problem into a set of bipartite matching subproblems

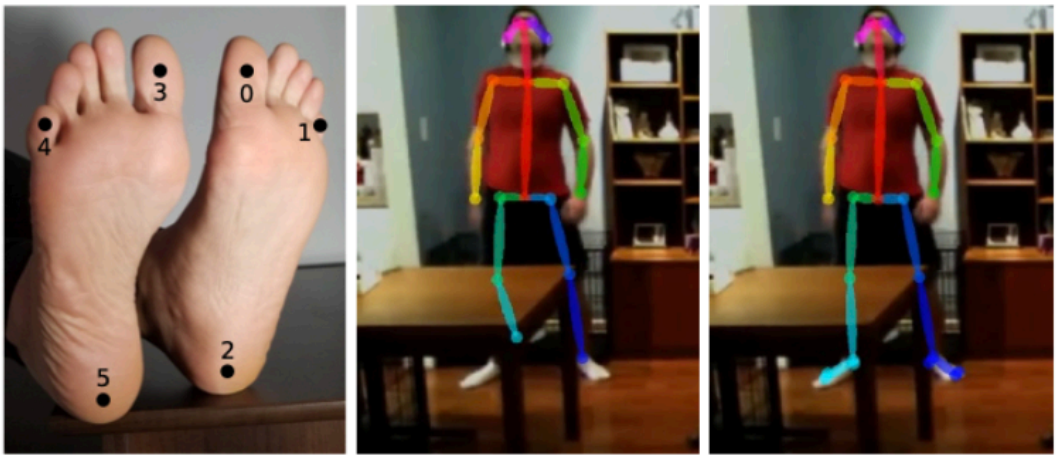
大致可以理解为先先通过图 b 的全连接结构获得最小边数生成树（图 c），然后将其转化为一个二分图匹配问题（图 d，带权，权值由上文提到的联系值公式获得）。

整体过程：



其他：

作者提出了脚部检测模型，这对身体其他肢体的检测可以起到帮助作用，如下图所示，中图没有使用该模型，脚踝的检测位置可能不太准确，右图使用了该模型，检测出脚部后隔着桌子也可以有效确定脚踝位置。


3
0
4
1
2
5