Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image

Introduction

SMPLify 能够全自动地捕获一张 2D 人像中的姿势信息与体型信息，从而生成对应的 3D 网格。首先，SMPLify 使用 CNN 来评估 2D 人像的姿势，并获取一组预测的 2D 关节点。之后，根据这组预测的 2D 关节点，SMPLify 能够计算出 3D 的 pose 信息与 shape 信息，生成对应的 SMPL 模型。

在先前，就已经存在不少关于如何根据 2D 关节点来评估 3D 姿势的讨论，但这些现有的方法都没有考虑人体的体型信息。与这些方法不一样，SMPLify 选用了 SMPL 模型来进行评估。SMPL 模型是一种参数化的蒙皮线性人体模型，它学习了大量的人体 pose 信息与 shape 信息。考虑到 SMPL 自带大量的 pose 信息与 shape 信息，使用 SMPL 的话，能够解决我们在根据 2D 关节点来评估 3D 姿势时，缺少 shape 信息的问题。

且外，大多数现有的方法都只能根据 2D 关节点来预测出 3D 的火柴人模型，且容易生成出不自然的 3D 姿势(主要是因为缺少深度信息，导致人体的不同部分不自然地相交在一起)。然而，使用传统的方法，想要解决人体模型不自然相交的问题 (interpenetration) 往往需要庞大的运算量。得益于 SMPL 是一种同时考虑 pose 信息与 shape 信息的模型，SMPLify 使用胶囊题来模拟人体的不同部分，然后通过定义与 pose 参数和 shape 参数有关的误差项来纠正人体姿势不自然的问题。

最后，SMPL 分别就男性和女性学习了两个不同的模版模型。因此，使用 SMPL 模型能让我们在评估不同性别的 2D 人像时获得更准确的结果。而为了应对无法获得性别信息的情况，SMPLify 还额外学习了一个性别无关的 SMPL 模版模型。

除去上述的使用 SMPL 带来的优势外，SMPLify 还使用 MoSh 从 CMU 数据集中学习了大量的人体姿势与体型的 prior。

Related Work

2D body joint(s) estimation：上文也提到了 SMPLify 需要先使用 CNN 来获得 2D 关节点。因为 SMPLify 需要用到的关节点是 LSP (Leeds Sports Pose) 关节点，所以我们需要使用基于 LSP 数据集训练得到的关节点预测器。作者推荐了 Convolutional Pose Machines 与 DeepCut (目前，存在 DeepCut 的改进版 — DeeperCut) 两个网络。我们需要将从 CNN 获得的关节点坐标与对应的预测自信度输入到 SMPLify 中。

SMPL：与传统的人体建模方法不一样，SMPL 考虑到了不同体型的人在做同一个姿势时也会存在差异。详细的内容可参考 SMPL 文档。

MoSh：传统的 marker-based motion capture (mocap，基于标记的运动捕获) 一直被诟病无法生产出生动的动画。而 motion and shape capture (MoSh) 则基于稀疏标记的数据并使用参数化的人体模型来同时评估 pose 信息与 shape 信息，它能够在不使用 3D scan 的情况下得到准确的 shape 信息。详细的内容可以参考 MoSh 论文。

Method

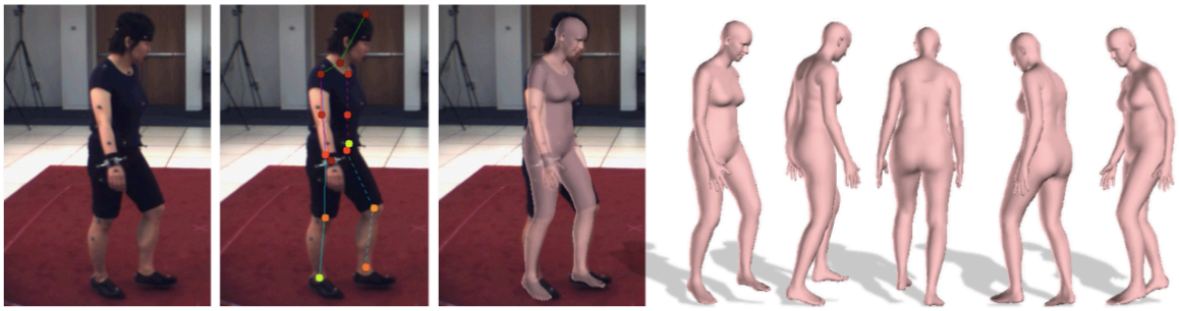


Fig. 2. System overview. Left to right: Given a single image, we use a CNN-based method to predict 2D joint locations (hot colors denote high confidence). We then fit a 3D body model to this, to estimate 3D body shape and pose. Here we show a fit on HumanEva [41], projected into the image and shown from different viewpoints.

首先，我们将一张单人人像输入到 DeepCut 或 CPM 中，并获得一组预测 2D 关节点 J_{est}。对于每一个 2D 关节点 J_i，都会有一个对应的预测自信度 w_i。然后，SMPLify 将这组 2D 关节点映射到 3D 上，并通过最小化误差项来拟合出 3D SMPL 人体模型。

SMPLify 的人体模型可以表示为 $M(\beta, \theta, \gamma)$，其中：β 表示 shape 参数；θ 表示 pose 参数；γ 表示相机的空间变换参数。SMPLify 的输出 $\mathcal{M}$ 是由 6890 个顶点构成的三角网格，且 β 与 θ 的定义与 SMPL 中的定义一致。SMPLify 包含了 3 个模版模型，使用 2000 个男性 registration 与 2000 个女性 registration 来训练出两个 gender-specific 模型 (粉色) 与一个 gender-neutral 模型 (蓝色)。

模型的姿势变换是通过 24 个关节点的骨骼蒙皮实现的，θ 是每个关节点的相对旋转角。最开始，我们能从模版模型中获得 3D 关节点与网格顶点的初始位置。然后，我们通过 β 计算出考虑人体体型后的关节点位置，记为 $J(\beta)$。之后，我们通过 θ 计算出考虑姿势变换后的关节点位置，记为 $R(J(\beta), \theta)$。我们还需要注意 LSP 关节点与 SMPL 关节点有一定的差异，无法构建出双射关系。因此，我们将 LSP 关节点与 SMPL 关节点中相似度最高的点联系在一起。最后，我们使用 perspective camera model 把 SMPL 关节点投影到图片上，我们把相机的投影参数记为 K。LSP 关节点与 SMPL 关节点标号的对应关系如下：

LSP	SMPL	LSP	SMPL
0 (right ankle)	8	1 (right knee)	5
2 (right hip)	2	3 (left hip)	1
4 (left knee)	4	5 (left ankle)	7
6 (right wrist)	21	7 (right elbow)	19
8 (right shoulder)	17	9 (left shoulder)	16
10 (left elbow)	18	11 (left wrist)	20
12 (neck)	-	13 (head top)	added (vertex id 411)

* 考虑到 LSP 与 SMPL 对 hips 的定义有较大的差异，且 SMPL 缺少对 neck 与 head top 的定义，我们有可能能构建更好的映射关系。

Approximating Bodies with Capsules:

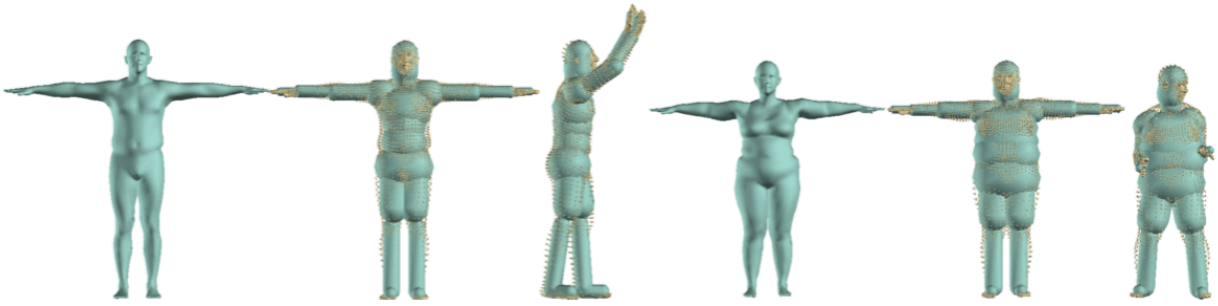


Fig. 3. Body shape approximation with capsules. Shown for two subjects. Left to right: original shape, shape approximated with capsules, capsules reposed. Yellow point clouds represent actual vertices of the model that is approximated.

上文也提及了，为了避免 interpenetration (即人体模型出现不自然的姿势，人的肢体不正常地相交在了一起)，我们往往需要进行额外的计算，而计算人体表面 (因为人体表面是复杂且非凹的) 的 interpenetration 的开销是非常大的，作者因此使用了 capsule (胶囊体) 来近似人体模型的不同部分。每个胶囊体有独立的半径与轴长。首先，我们手动初始化一组胶囊体来表示 SMPL 模版模型中除手指与脚趾外的 20 个部分。然后，我们使用梯度下降的方式最小化胶囊体表面与模型表面的双向距离。之后，我们使用交差验证脊回归 (cross-validated ridge regression) 来学习一个从体型参数 到胶囊体参数的线性回归子。最后，我们将回归子的输出作为胶囊体的初始化，再重复一遍上述流程。

* 通过改进学习的方法，我们有可能能学习到更好的回归子。

Objective Function:

首先，我们给出需要最小化的目标函数 $E(\beta, \theta)$，它是 5 个不同误差项的加权和。

$$E_J(\boldsymbol{\beta}, \boldsymbol{\theta}; K, J_{\text{est}}) + \lambda_\theta E_\theta(\boldsymbol{\theta}) + \lambda_a E_a(\boldsymbol{\theta}) + \lambda_{sp} E_{sp}(\boldsymbol{\theta}; \boldsymbol{\beta}) + \lambda_\beta E_\beta(\boldsymbol{\beta})$$

我们使用关节点误差项 (joint-based data term) 来惩罚预测关节点与 2D 关节点的加权距离和。其中，Π_K 表示预测 3D 关节点根据摄像机投影参数 K 得到的 2D 投影关系。根据不同 2D 关节点的预测自信度 w_i，我们在误差项中为其赋予不同的比重。最后，我们使用 Geman-McClure 范数来代表预测关节点与 2D 关节点的距离，借此来改善处理模糊输入图片时的表现。

$$E_J(\boldsymbol{\beta}, \boldsymbol{\theta}; K, J_{\text{est}}) = \sum_{\text{joint } i} w_i \rho(\Pi_K(R_\theta(J(\boldsymbol{\beta})_i)) - J_{\text{est},i})$$

为了解决 interpenetration 的问题，我们需要引入一个基于先验知识的误差项。这里的 i 代表与膝盖弯曲、肘部弯曲有关的姿势节点。这里的 θ 代表 rotation 的角度，正常情况下，θ 为负值，则 $0 < \exp(\theta) < 1$；若 θ 为正值，意味着发生了不自然的扭转，$\exp(\theta) \gg 1$。需要注意的是，θ 在不发生旋转时等于 0。

$$E_a(\boldsymbol{\theta}) = \sum_i \exp(\boldsymbol{\theta}_i)$$

为了进一步淘汰掉不正常的姿势，我们可以预先为模型设定一些 favorable pose (可以理解成较为常见的姿势)。作者在 CMU marker dataset 上通过 MoSh 来获得一系列的 SMPL 模型，然后用这些模型来建立若干个高斯分布 (即混合高斯模型)。在公式中，g_j 表示 $N = 8$ 共 8 个高斯分布中的第 j 个的权值，$\mathcal{N}()$ 表示概率密度函数。正常情况下，我们需要计算所有高斯分布的加权概率和的负对数，但这种做法在运算上的开销非常大，我们可以用所有高斯分布中最大的加权概率来代替所有高斯分布的加权概率和。因此，我们引入一个常量 c 来抵消这种近似带来的影响。

$$E_\theta(\boldsymbol{\theta}) \equiv -\log \sum_j (g_j \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\theta,j}, \Sigma_{\theta,j})) \approx -\log(\max_j (c g_j \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\theta,j}, \Sigma_{\theta,j})))$$
$$= \min_j (-\log(c g_j \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\theta,j}, \Sigma_{\theta,j})))$$

* 更加常见的姿势通常会有较大的概率密度函数值，因此，其负对数会更接近 0 。

之前我们提到使用 capsule 来近似人体部分，并以此来改善 interpenetration 的问题。我们进一步将 capsule 近似为 sphere (以便于计算胶囊体的体积)，这里的 $C(\theta, \beta)$ 代表胶囊的中心坐标，$r(\beta)$ 代表胶囊的半径，$\sigma(\beta)$ 等于 $r(\beta)/3$。$I(i)$ 则是对于第 i 个球体，与其不相容的球体的集合。$||C_i(\theta, \beta)$ - $C_j(\theta, \beta)||^2$ 表示两个球体的球心距，$\sigma_i(\beta)^2 + \sigma_j(\beta)^2$ 则用于近似两个球体的半径和。

$$E_{sp}(\boldsymbol{\theta}; \boldsymbol{\beta}) = \sum_i \sum_{j \in I(i)} \exp\left(\frac{||C_i(\boldsymbol{\theta}, \boldsymbol{\beta}) - C_j(\boldsymbol{\theta}, \boldsymbol{\beta})||^2}{\sigma_i^2(\boldsymbol{\beta}) + \sigma_j^2(\boldsymbol{\beta})} \right)$$

* 通过比较球心距和半径和，我们可以判断两个球体的相对位置关系。当球心距大于等于半径和，代表两个球相切或不相交；当球心距小于半径和，代表两个球相交或包含。所以，在我看来，正常情况下应该选择最小化半径和与球心距的比值。在这个疑惑下，我查看代码并发现了论文存在的纰漏。$||C_i(\theta, \beta)$ - $C_j(\theta, \beta)||^2$ 不是表示两个球体的球心的二范数，而是二范数的平方；且正确公式应该为 $\sum\sum \exp((||C_i(\theta, \beta)$ - $C_j(\theta, \beta)||^2) \div (\sigma_i(\beta)^2 + \sigma_j(\beta)^2))$。除此之外，我暂时没能理解文章使用 isotropic Gaussian 的意义 (即令 $\sigma(\beta)$ 等于 $r(\beta)/3$)。我们需要注意，这个误差项仅能用于优化姿势参数，因为考虑体型参数的话，这个误差项会令模型趋向于纤细。由此可见，我们有较大的空间来改进这一个误差项。

最后，作者还定义了一个体型误差项，其中，$\sum_\beta^{-1}$ 表示 SMPL 训练过程中 PCA 获得的对角线矩阵。

$$E_\beta(\boldsymbol{\beta}) = \boldsymbol{\beta}^T \Sigma_\beta^{-1} \boldsymbol{\beta}$$

* 我在查看代码后发现，这个误差项实指 β 的和 (根据 SMPL 模型的定义，β 代表与模版模型 shape 的 offset，所以 β 等于 0 时，模型将拥有 mean shape：即惩罚 β 能让生成的模型更接近 mean shape)，所以这条公式在表达上貌似有点错误。 (据我所知，PCA 的对角线矩阵不是这样用的。我没法理解这条公式的含义，或许其实际意义与 β 的和是一样的。最后，我认为惩罚 β 的和是完全合理的。)

Optimization:

摄像机的平移 γ 与人体的朝向是未知的，但在估计出这两个变量前，我们需要先获得摄像机的焦距。在这里，SMPLify 把摄像机的焦距固定为 5000。然后，我们假设照片中的人拥有 mean shape。通过与

SMPL 模版模型作对比，比较 2D 人像与模版模型的躯干长度，从而通过相似三角形初始化出摄像机的平移 γ (此时，摄像机只在 z 轴方向有位移)。接下来，我们通过计算两个 2D 肩部关节点的距离来判断人像是正面朝向摄像机还是侧身朝向摄像机 (正面朝向时，肩部关节点的距离会较大；侧身朝向时，肩部关节点的距离会较小)。考虑到初始化 γ 的值未必是准确的 (人物未必拥有 mean shape)，我们在获得人体朝向后，我们固定 β，使用上文提到的 $E_J()$ 来初步优化 γ 与 θ。最后，在完成上述操作后，我们再使用 Powells dogleg method 来优化 objective function，获得需要的三角网格。需要注意，作者在处理侧身朝向的照片时，考虑了朝左和朝右两种情况，并取较优者作为最终结果。且外，作者在训练过程中逐步减小 objective function 中 λ_θ 与 λ_β 的值，防止陷入局部最优 (因为这两个权值对应的误差项都是基于先验知识的，如果这两个权值过大，拟合出来的模型将会趋近于模版)。

* 首先，SMPLify 把摄像机的焦距固定为 5000，而这样的假设在实际情况中显然是不可靠。因此，当实际焦距与 5000 有较大差距时，SMPLify 很可能无法获得较好的表现。但因为我们实际拍照时，不同相机的焦距不会有太大差异，所以这一定程度上减少了固定焦距的影响。我们也可以在拟合模型时尝试几个比较常见的焦距，然后选取误差值最小的一个作为最终结果，从而改进 SMPLify 的表现。我们也能发现，如果无法通过 2D 关节点获得准确的躯干长度的话，我们就无法估计出 γ 的值。而且在计算 γ 的过程中，我们始终假设人物是拥有 mean shape 的，这或多或少会有一点误差 (我认为影响不会太大)。最后，作者在处理侧身照时，分别考虑了朝左和朝右两种情况，但我认为 SMPL 模型在学习过程中已经通过 blend shape 避免了不自然的姿势，因此，即便我们不分别考虑朝左和朝右，我们大概也能拟合出朝向正确的模型。

Evaluation

考虑到实际情况中，一幅 2D 图片很少会带有一个 ground truth 的 3D 模型，作者在此同时使用了 synthetic data 与 real data 来进行评估。在 synthetic data 部分中，作者先从一系列 SMPL 模型中获取到 2D 节点，并对这些 synthetic data 添加随机的高斯噪声，然后以 synthetic data 为输入，通过上述方法对模型进行评估。且外，作者还在已知正确的 pose 参数的情况下，尝试用尽量少的关节点来拟合模型。这部分的结果如下：

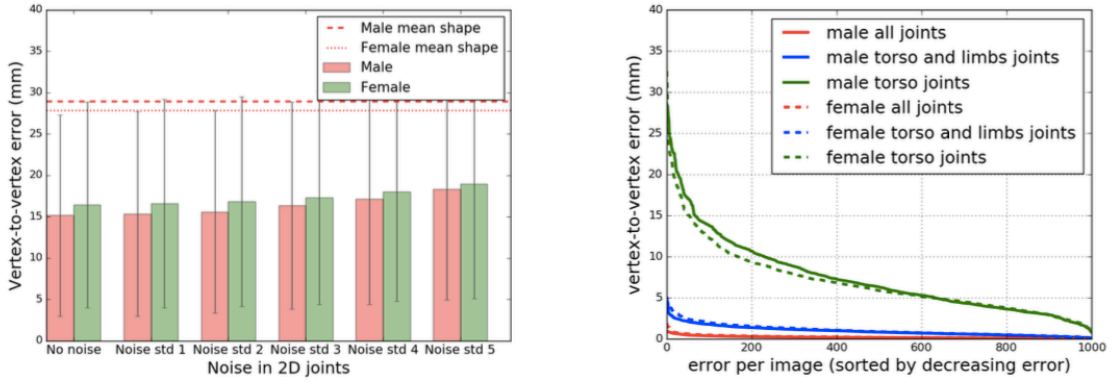


Fig. 4. Evaluation on synthetic data. Left: Mean vertex-to-vertex Euclidean error between the estimated and true shape in a canonical pose, when Gaussian noise is added to 2D joints. Dashed and dotted lines represent the error obtained by guessing the mean shape for males and females, respectively. Right: Error between estimated and true shape when considering only a subset of joints during fitting.

在 real data 部分中，作者分别在 HumanEva-I dataset 与 Human3.6M. dataset 上进行 quantitative evaluation，其中，这两个数据集都是自带 ground truth 的。这部分没有什么分析的空间，具体结果如下。

	Walking			Boxing			Mean	Median
Method:	S1	S2	S3	S1	S2	S3		
$E_\beta + E_J + E_{\theta'}$	98.4	79.6	117.8	105.9	98.5	122.5	104.1	82.3
$E_\beta + E_J + E_{\theta'} + E_{sp}$	97.9	79.4	116.0	105.8	98.5	122.3	103.7	82.3
SMPLify	**73.3**	**59.0**	**99.4**	**82.1**	**79.2**	**87.2**	**79.9**	**61.9**

Table 2. HumanEva-I ablation study. 3D joint errors in mm. The first row drops the interpenetration term and replaces the pose prior with a uni-modal prior. The second row keeps the uni-modal pose prior but adds the interpenetration penalty. The third row shows the proposed SMPLify model.

	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing	Purchases	Sit
Akhter & Black [4]	199b.2	177.6	161.8	197.8	176.2	186.5	195.4	167.3	160.7
Ramakrishna et al. [39]	137.4	149.3	141.6	154.3	157.7	158.9	141.8	158.1	168.6
Zhou et al. [58]	99.7	95.8	87.9	116.8	108.3	107.3	93.5	95.3	109.1
SMPLify	**62.0**	**60.2**	**67.8**	**76.5**	**92.1**	**77.0**	**73.0**	**75.3**	**100.3**
	SitDown	Smoking	Waiting	WalkDog	Walk	WalkTogether	Mean	Median	
Akhter & Black [4]	173.7	177.8	181.9	176.2	198.6	192.7	181.1	158.1	
Ramakrishna et al. [39]	175.6	160.4	161.7	150.0	174.8	150.2	157.3	136.8	
Zhou et al. [58]	137.5	106.0	102.2	106.5	110.4	115.2	106.7	90.0	
SMPLify	**137.3**	**83.4**	**77.3**	**79.7**	**86.8**	**81.7**	**82.3**	**69.3**	

Table 3. Human 3.6M. 3D joint errors in mm.

至于 qualitative evaluation，作者在 LSP dataset 上进行了测试。SMPLify 不仅在一些简单姿势上能获得明显优于其他模型的效果，SMPLify 在处理复杂的运动姿势时依然也能给出很好的表现。

Fig. 8. Qualitative comparison. From top to bottom: Input image. Akhter & Black [4]. Ramakrishna et al. [39]. Zhou et al. [58]. SMPLify.

Fig. 6. Leeds Sports Dataset. Each sub-image shows the original image with the 2D joints fit by the CNN. To the right of that is our estimated 3D pose and shape and the model seen from another view. The top row shows examples using the gender-neutral body model; the bottom row show fits using the gender-specific models.

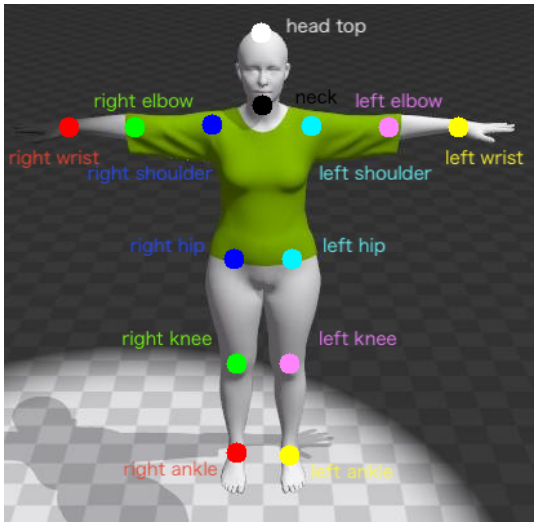
上述内容都不需要我们认真关注，我们只用把重点放在能否将 SMPLify 应用于虚拟试衣上。SMPLify 在 LSP dataset 上运行时，遇到了下列的失效情况：罕见姿势导致的错误预测 (这与 SMPL 模型的学习过程有关，不太好修正这个问题)；处理多人合照时的错误预测。

Fig. 7. LSP Failure cases. Some representative failure cases: misplaced limbs, limbs matched with the limbs of other people, depth ambiguities.

在处理虚拟试衣问题时，我们不会遇到多人合照的问题，但是遇到罕见姿势时可能会出现问题。不过，我们显然不会遇到像 Fig. 7(1) 中的手和脚高度重叠的问题，也不会遇到像 Fig. 7(2) 中关节点旋转角为正的问题。在上文中，我已经讨论了很多 SMPLify 存在的不足，而对于虚拟试衣来说，它最严重的问题在于无法处理一些特殊的体型。因为 SMPL 模型学习了各种体型的 blend shape，所以我们拟合出来的模型是趋向于拥有均匀的体型的。如果我们想尝试处理一些拥有不均匀体型的人像 (比如说孕妇之类的)，那么生成的模型肯定会与实际有较大的误差。

· Appendix

如果我们想要在自己的数据集上运行 SMPLify，简要的流程如下。



1) 将 2D 人像输入至使用 LSP dataset 训练得到的 2D joints estimator 中 (如 DeepCut、CPM 等，现今可能有性能更好的 estimator)，获得一组预测关节点与对应的预测自信度。

2) 我们以 DeepCut 举例：预测关节点与对应的预测自信度被储存为一个 numpy 数组 (或输出成 npz 文件)。这个 numpy 数组的大小为 5 × 13，对应的关节点编号已在上文中标明，其中，第一维中数据的含义如下：position x、position y、CNN confidence、CNN offset vector x、CNN offset vector y。因此，在 SMPLify 中我们只用关心第一维的前三组数据即可。

3) 将运行 DeepCut 的得到的数据与原始人像图片作为参数，传入 SMPLify 的 run_single_fit() 函数。之后，我们可以获得一个包括以下 key 值的 dict 类型的变量：cam_t、f、pose、betas。cam_t 是摄像机的平移；f 是摄像机的焦距；pose 是 SMPL 模型的 pose 参数；betas 是 SMPL 模型的 shape 参数。如果我们只想实现 3D 上的虚拟试衣，那么我们没必要关注 cam_t 与 f；如果我们想要在实现完 3D 上的虚拟试衣后把试衣结果贴回原图像上，我们需要在最后使用 cam_t 与 f 来求解摄像机。

4) 当我们想要通过 pose 与 betas 还原出对应的 SMPL 模型时，我们需要先调用 SMPL 的 load_model() 函数读取对应性别的 SMPL 模版模型，然后把 model 的 pose 与 betas 参数修改成我们先前得到的值，最后输出成 obj 文件。(以下代码与我们的需求略有区别)

```
from smpl_webuser.serialization import load_model
import numpy as np

## Load SMPL model (here we load the female model)
## Make sure path is correct
m = load_model( '../../models/basicModel_m_lbs_10_207_0_v1.0.0.pkl' )

## Assign random pose and shape parameters
m.pose[:] = np.random.rand(m.pose.size) * .2
m.betas[:] = np.random.rand(m.betas.size) * .03

## Write to an .obj file
outmesh_path = './hello_smpl.obj'
with open( outmesh_path, 'w') as fp:
    for v in m.r:
        fp.write( 'v %f %f %f\n' % ( v[0], v[1], v[2]) )

    for f in m.f+1: # Faces are 1-based, not 0-based in obj files
        fp.write( 'f %d %d %d\n' %  (f[0], f[1], f[2]) )

## Print message
print ('..Output mesh saved to: ', outmesh_path)
```

5) 如果我们还想把试衣结果贴回原图像上，我们只需调用 SMPLify 的 render_model() 函数即可。

```
def render_model(verts, faces, w, h, cam, near=0.5, far=25, img=None):
    rn = _create_renderer(
        w=w, h=h, near=near, far=far, rt=cam.rt, t=cam.t, f=cam.f, c=cam.c)
    # Uses img as background, otherwise white background.
    if img is not None:
        rn.background_image = img / 255. if img.max() > 1 else img

    imtmp = simple_renderer(rn, verts, faces)

    # If white bg, make transparent.
    if img is None:
        imtmp = get_alpha(imtmp)

    return imtmp
```

总的来说，如果我们想要把 SMPLify 应用于虚拟试衣上，我们只能借助它生成 obj 的模型。因为 SMPL 的 shape 参数 (即 betas) 并没有明确的实际意义，只是通过 PCA 提取的特征。且外，在完成这篇文档后，我发现了 Pavlakos, Georgios & Zhu, Luyang & Zhou, Xiaowei & Daniilidis, Kostas. (2018). Learning to Estimate 3D Human Pose and Shape from a Single Color Image. 这篇论文。与 SMPLify 不一样，这篇论文没有使用回归的方式来拟合出 SMPL 模型，而是训练了一个 CNN。

这篇论文被收录于 CVPR 2018，并表示它的性能要优于 SMPLify。由于时间关系，我没有办法去仔细分析它的研究成果。但鉴于它的输出也是一个 SMPL 模型，我们或许能用它来代替 SMPLify (具体流程肯定会有区别)。