

Photo Wake-Up: 3D Character Animation from a Single Photo

Патакин Николай

Высшая школа экономики

19 апреля 2019 г.

Какая задача решается авторами?

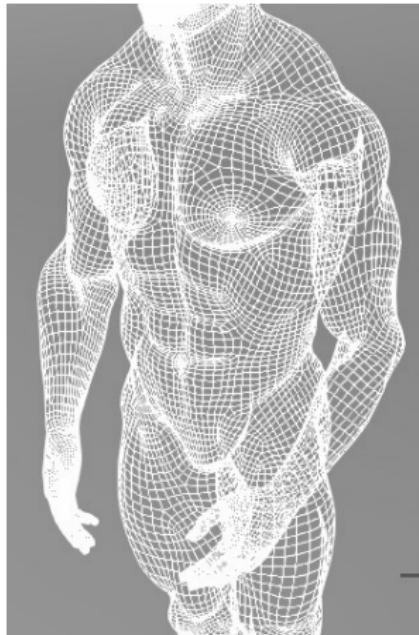


Input Photo



Photo Animation

Введение. Примитивы 3D графики



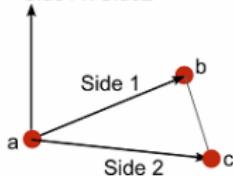
3D модель представляется в виде набора вершин \mathcal{V} и набора индексов вершин, входящих в состав треугольников \mathcal{F}

$$\mathcal{V} \in \mathbb{R}^{n \times 3}$$

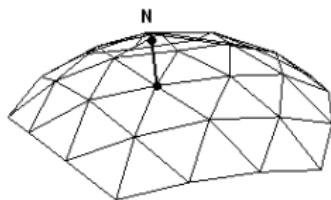
$$\mathcal{F} \in \mathbb{N}^{m \times 3}$$

Введение. Примитивы 3D графики

$$\text{Normal} = \text{Side1} \times \text{Side2}$$

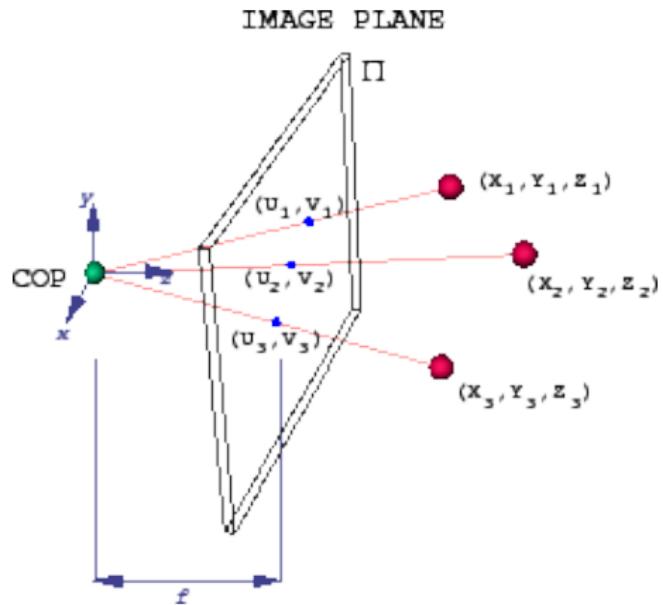


Нормаль к треугольнику может быть рассчитана, как векторное произведение векторов, задающих плоскость треугольника



Нормаль к вершине рассчитывается как среднее между нормалями, прилегающими к ней треугольников

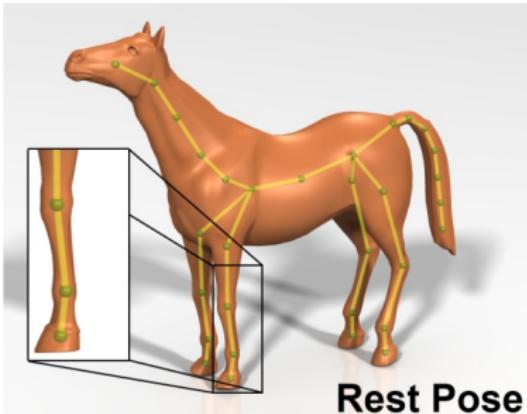
Введение. Как устроена камера?



Простейшую камеру можно охарактеризовать с помощью её:

- положения C_p
- направления обзора C_d
- фокусного расстояния f
- соотношения сторон проекционной плоскости

Введение. Скелетная анимация



Для анимации произвольной модели необходима дополнительная информация:

\mathcal{J} - позиции joint-ов(точек скелета, представляющегося в виде дерева), $\mathcal{J} \in \mathbb{R}^{k \times 3}$

Как закодировать изменение позы?

Для каждого joint-а: $\theta_i \in \mathbb{R}^3$, $\frac{\theta_i}{\|\theta_i\|}$ – нормализованная ось, вокруг которой происходит вращение, $\|\theta_i\|$ – угол вращения.

θ_i однозначно кодирует матрицу трансформации $R_i \in \mathbb{R}^{4 \times 4}$ – вращение вдоль оси $\frac{\theta_i}{\|\theta_i\|}$ на величину $\|\theta_i\|$ вокруг точки \mathcal{J}_i

R_i – локальные матрицы трансформации. Для каждого joint-а нужно получить глобальную матрицу: $\hat{R}_i = R_{p_0} \cdot R_{p_1} \cdots R_i$, где p_0, p_1, \dots, i – путь от корня до вершины в дереве скелета

Как трансформировать вершины?

Нужна матрица весов - $\mathcal{W} \in \mathbb{R}^{n \times k}$
Тогда координаты новой вершины:

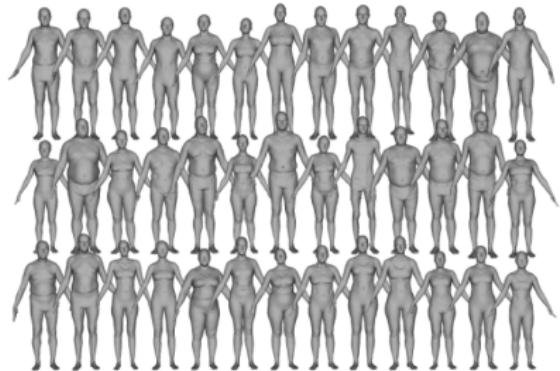


$$v'_i = \left(\sum_{j=1}^k w_{ij} \hat{R}_j \right) \cdot v_i$$

Требования к весам:

- $w_{ij} \geq 0$
- $\sum_j w_{ij} = 1$
- Дальше вершина от точки скелета \Rightarrow меньше вес точки скелета

Пространство формы тела человека



Параметрическая модель тела человека

Имеется множество сканов людей \mathcal{S} . Имеется шаблонная модель тела человека \mathcal{T} .

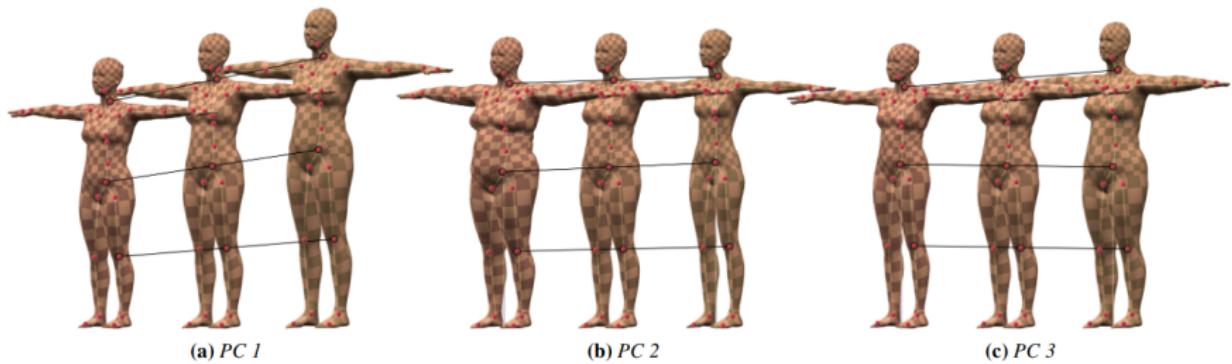
Для каждого скана шаблон \mathcal{T} деформируется $\Rightarrow \tilde{\mathcal{T}}_i$ так, чтобы как можно точнее отражать скан \mathcal{S}_i . Выполняем PCA-разложение на множество деформированных шаблонов $\tilde{\mathcal{T}}$. Получаем:

- Усредненную форму тела человека $\bar{\mathcal{T}}$
- Собственные вектора $d_i \in \mathbb{R}^{n \times 3}$. Оставляем несколько первых(10-20 компонент)
- Низкоразмерное представление формы человека:

$$\mathcal{V} = \bar{\mathcal{T}} + \sum_{i=0}^{|d|} \beta_i d_i$$

β – вектор параметров формы

PCA-компоненты параметрической модели



Формулировка параметрической модели SMPL

Параметрическая модель объясняет пространства:

- Формы. Элемент пространства определяется параметрами β
- Позы, задаваемой параметрами $\vec{\theta}$. Модель, полученная после применения параметров формы, трансформируется алгоритмом скелетной анимации LBS

Для модели SMPL тренируются параметры $\bar{T}, \mathcal{S}, \mathcal{W}, \mathcal{J}$:

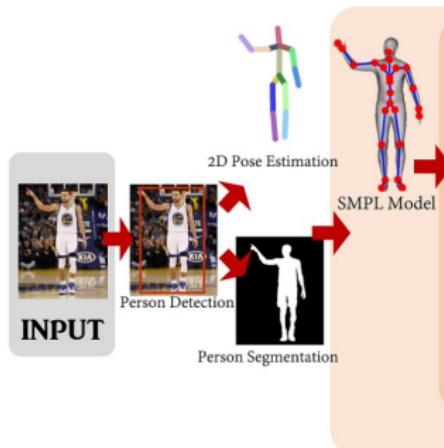
- \bar{T}, \mathcal{S} – усредненная модель человека и собственные вектора, полученные PCA разложением
- \mathcal{W} – матрица весов для алгоритма скелетной анимации
- $\mathcal{J}(\beta)$ – линейный регрессор положения точек скелета(линейный от координат вершин!)

Применения параметрической модели

- Генерация разнообразных персонажей для игр, фильмов и т.д.
- Автоматическая анимация существующих 3D-моделей человека
- Автоматические обмеры человека для производства одежды
- Дизайн эргономики
- Основа для других алгоритмов компьютерного зрения – задачи восстановления 3D-модели человека по частичным данным(сенсор глубины, видео, обычное цветное изображение)

Алгоритм, предлагаемый авторами

Для начала:



- ➊ Произвести детекцию и сегментацию силуэта человека от фона
- ➋ Найти на изображении точки, соответствующие проекции joint-ов скелета
- ➌ С использованием позиций joint-ов и силуэта человека восстановить параметры SMPL, соответствующие изображению

Сегментация силуэта человека

Использована архитектура Mask-RCNN, предназначенная для детекции, классификации и сегментации объектов на изображении



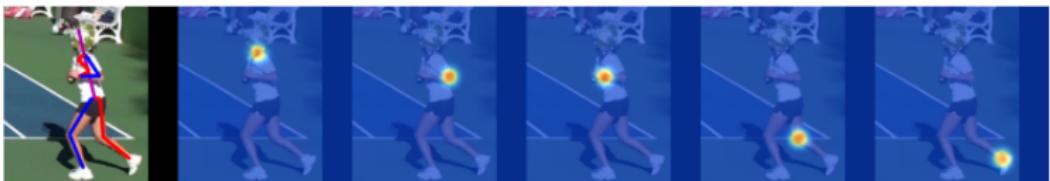
2D Pose Estimation

Для каждой точки скелета нейросеть предсказывает координаты пикселя, в котором эта точка находится на изображении, а также степень уверенности в детекции



2D Pose Estimation. Секреты успеха

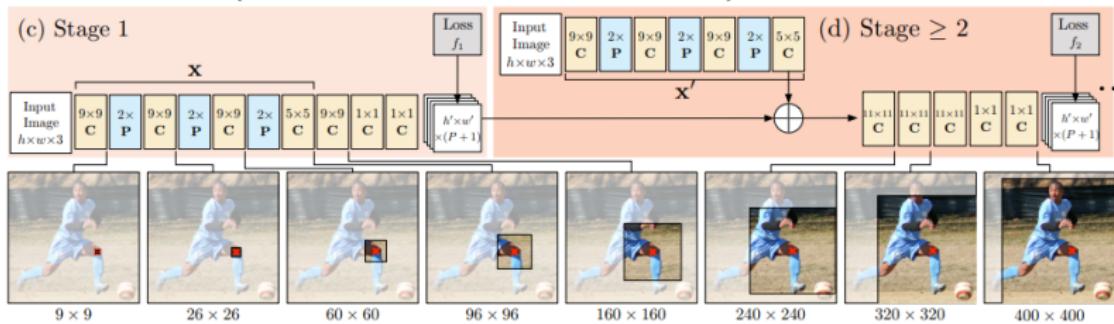
- Предсказывать координаты точек - плохая идея! Гораздо лучше работает предсказание *heatmap*-ов (карт вероятности появления ключевой точки в конкретном пикселе):



- Отдельные нейросети на каждую ключевую точку - плохо, теряется пространственная связь разных точек. Выход нейросети стоит делать - тензор $W \times H \times K$, где W, H - размеры изображений, K - число точек скелета

2D Pose Estimation. Секреты успеха

- Использование сверток большого размера очень хорошо увеличивает "область восприятия" нейросети
- Стэкать сетки, исправляющие ошибки предыдущих(Convolutional Pose Machine):



Восстановление параметров SMPL по изображению

Авторы данной статьи предлагают использовать здесь метод Bogo et al. 2016 без изменений. Метод принимает на вход:

- Позиции джоинтов на изображении(2D Pose Estimation)
- Сегментированный силуэт человека

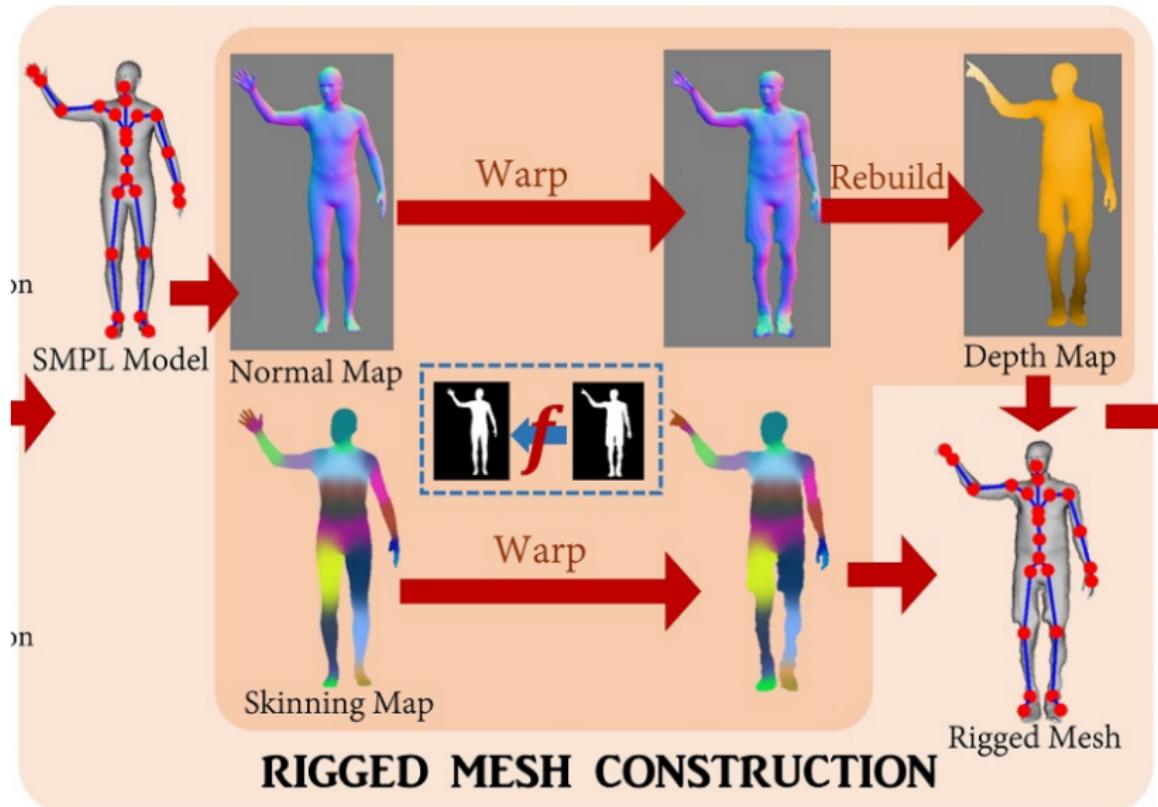
Алгоритм восстанавливает следующие параметры:

- Параметры камеры(положение, направление и т.д.)
- Параметры формы SMPL модели
- Параметры позы SMPL модели

Пример реконструкции параметров SMPL



Преобразование силуэтов



Преобразование силуэтов

Пусть x - пиксель изображения(его координаты).

$$S(x) = \begin{cases} 1, & x \in \text{human_mask} \\ 0, & \text{else} \end{cases}$$

Проектируем SMPL, используя полученные параметры камеры, получаем маску S_{SMPL}

- $Z(x) \in \mathbb{R}$ – карта глубины(расстояния до ближайшей точки от камеры)
- $N(x) \in \mathbb{R}^3$ – карта нормалей
- $W(x) \in \mathbb{R}^{24}$ – карта весов скиннинга

Преобразование силуэтов

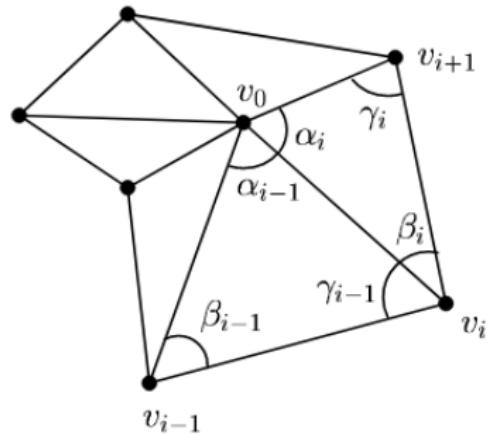
$$S(x) = S_{SMPL}(f(x)), \quad f(x) - ?$$

Пусть $p_i \in \partial S$, $p_i^{SMPL} \in \partial S_{SMPL}$ – точки принадлежащие границе силуэта сегментационной маски изображения и SMPL соответственно. Установим соответствие между точками границ контуров:

$$p_i \rightarrow p_{\phi[i]}^{SMPL}$$

Преобразование силуэтов

Представляем произвольную вершину $x \in S$ как выпуклую комбинацию $p_i \in \partial S$. Делаем это с помощью Mean Value Coordinates:



$$\sum_{i=1}^k \lambda_i p_i = x, \quad \sum \lambda_i = 1$$

$$\lambda_i = \frac{w_i}{\sum w_j}, \quad w_i = \frac{\cot \gamma_{i-1} + \cot \beta_i}{||v_i - x||^2}$$

$$\Rightarrow f(x) = \sum_{i=1}^k \lambda_i p_{\phi[i]}^{SMPL}$$

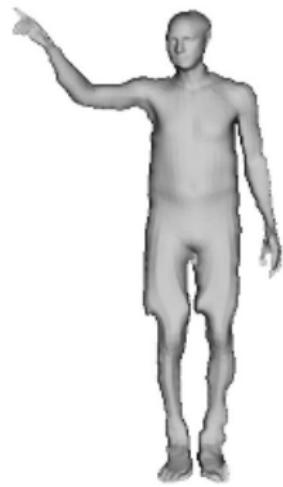
Как выбрать соответствие вершин силуэтов?

$$\arg \min_{\phi[0] \cdots \phi[m-1]} \sum_{i=0}^{m-1} ||p_i - p_{\phi[i]}^{SMP^L}|| + T(\phi[i], \phi[i+1])$$

где

$$T(\phi[i], \phi[i+1]) = \begin{cases} 1, & \text{if } 0 \leq \phi[i+1] - \phi[i] \leq 32 \\ \infty, & \text{else} \end{cases}$$

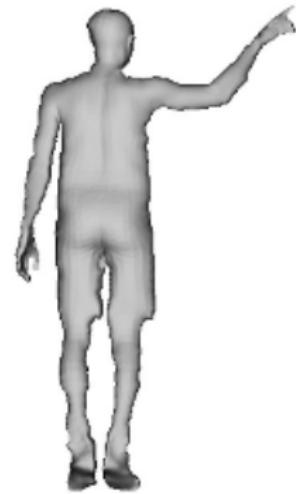
Эффективно решается динамическим программированием!



(a) front mesh

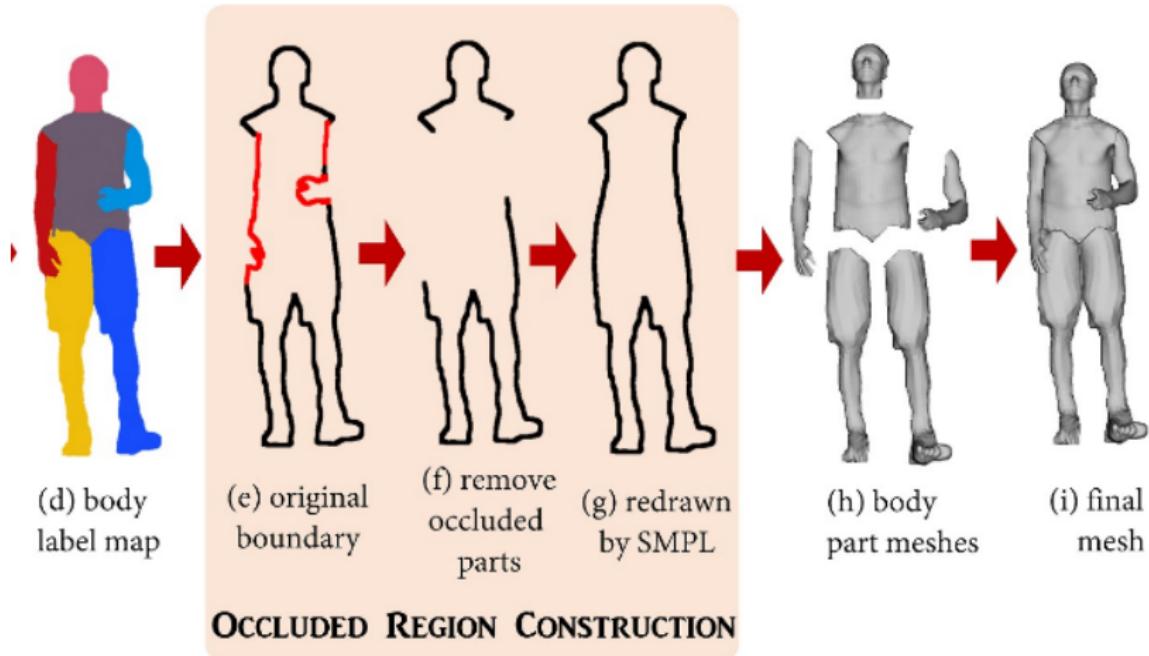


(b) side view



(c) back mesh

Как обрабатывать пересечения рук и тела?



- В докладе была рассказана концепция параметрической модели человека и одном из её применений
- Для задачи анимации персонажей по единственному цветному изображению авторы предложили честный метод, работающий с применением 3D реконструкции
- В отличии от существующих работ, которые предлагают только реконструкцию при помощи параметрической модели, авторы придумали метод уточнения силуэтов
- Авторами статьи решена проблема частичного перекрывания силуэта частями тела

Список литературы

- ① Chung-Yi Weng, Brian Curless, Ira Kemelmacher-Shlizerman, Photo Wake-Up: 3D Character Animation from a Single Photo
- ② Matthew Loper, Naureen Mahmood, Javier Romero, SMPL: A Skinned Multi-Person Linear Model
- ③ Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape
- ④ Shih-En Wei, Varun Ramakrishna, Takeo Kanade, Convolutional Pose Machines
- ⑤ Michael S. Floater, Mean Value Coordinates