Robust Estimation of 3D Human Poses from a Single Image

阅读报告

张淦淦

2020 年 6 月 27 日

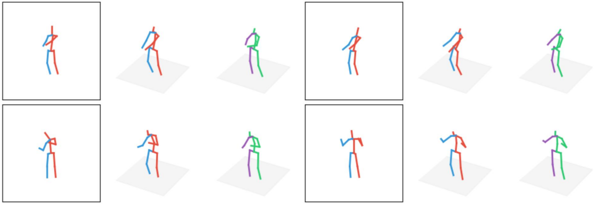
目录

目标

单幅图像3D人体姿态估计,可以分为两个阶段进行.

- 第一个是由图像得到2D关键点信息,可分为单人2D关键点检测,多人2D关键点检测(单人2D关键点检测研究已经比较成熟,相关方法如stacked hourglass(2017))
- 第二个阶段是阶段是从已知2D信息估计出3D姿势信息.

本文讨论的是已知单人2D关键点信息估计出3D姿势信息的方法.



基本模型

假设n表示关键点的数量,2D姿态与3D姿态分别用$x \in R^{2n}$和$y \in R^{3n}$表示,对2D和3D数据要进行以均值为坐标原点的处理后,设$M_0 = \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} \in R^{2\times 3}$表示弱透视模型相机矩阵,其中$m_1^T m_2 = 0$,则

$$x = My$$

在以往的研究中显示,人体姿态很适合用在稀疏空间中表示[?],设$B = b_1, \ldots, B_k \in R^{3n\times k}$是通过字典学习的方法学习到的字典[?](实际中用spams库实现,k是自己设定的字典长度,本次实验设计为200),$\alpha \in R^{k\times 1}$是对应的稀疏向量.(实现中μ为0)

$$y = B\alpha + \mu$$

优化问题的目标函数

根据2D关键点估计3D姿态可以转化为,一个优化问题

$$\min_{\alpha} ||x - M(B\alpha + \mu)||_1 + \theta||\alpha||_1$$

加入约束,设肢体各个部分的长度分别为L_i,则有

$$||C_i y||_2^2 = L_i$$

$$C_i = E_{i1} - E_{i2}$$

$$E_j = [0, \ldots, I, \ldots, 0] \in R^{3\times 3n}$$

可总结为,在相机参数M已知的情况下

$$\begin{aligned} \min_{\alpha} \quad & ||x - M(B\alpha + \mu)||_1 + \theta||\alpha||_1 \\ s.t. \quad & ||C_i(B\alpha + \mu)||_2^2 = L_i \end{aligned}$$

已知y,而相机M参数不知时,通过以下优化问题求相机参数

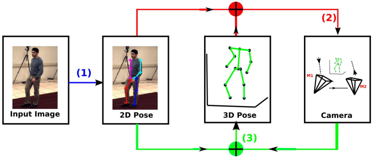
$$\begin{aligned} \min_{m_1, m_2} \quad & ||x - \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} y||_1 \\ s.t. \quad & m_1^T m_2 = 0 \end{aligned}$$

求3D姿态y需要相机参数M,而求M时需要y,形成依赖环路,此时用循环迭代求解的方式解决.

总结

$$
\begin{aligned}
\min_{\alpha} \quad & ||x - M(B\alpha + \mu)||_1 + \theta||\alpha||_1 \\
s.t. \quad & ||C_i(B\alpha + \mu)||_2^2 = L_i
\end{aligned} \tag{1}
$$

$$
\begin{aligned}
\min_{m_1, m_2} \quad & ||X - \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} Y||_1 \\
s.t. \quad & m_1^T m_2 = 0
\end{aligned} \tag{2}
$$



1. 对y进行预处理.通过字典学习得到B
2. 设定初始值,$y_k = y_0$
3. 将y_k带入(2)求得M_k
4. 将M_k带入(1),求得y_{k+1}
5. 3,4循环直到$||M_k - M_{k-1}||_2 < \sigma_1$和$||y_k - y_{k-1}||_2 < \sigma_1$

ADMM算法补充

ADMM算法可以解决以下形式的优化问题

$$\begin{aligned} min \quad & f(x) + g(z) \\ s.t. \quad & Ax + Bz = c \end{aligned}$$

其中f和g都是凸函数,其增强拉格朗日方程如下

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}||Ax + Bz - c||_2^2$$

ADMM算法迭代方程

$$\begin{aligned} x^{k+1} =& \underset{x}{argmin}\, L_\rho(x, z^k, y^k) \\ z^{k+1} =& \underset{z}{argmin}\, L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} =& y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{aligned}$$

可变形为

$$\begin{aligned} x^{k+1} =& \underset{x}{argmin} \left(f(x) + \frac{\beta}{2}||Ax + Bz^k - c + \lambda^k/\beta||_2^2 \right) \\ z^{k+1} =& \underset{z}{argmin} \left(g(z) + \frac{\beta}{2}||Ax^{k+1} + Bz - c + \lambda^k/\beta||_2^2 \right) \\ \lambda^{k+1} =& \lambda_k + \beta[Ax^{k+1} + Bz^{k+1} - c] \end{aligned} \tag{3}$$

2D到3D目标方程求解

转换形式,方便借用ADMM方法求解

$$\begin{aligned} \min_{\alpha,\beta,\gamma} \quad & ||\gamma||_1 + \theta||\beta||_1 \\ s.t. \quad & \gamma = x - M(B\alpha + \mu) \\ & \alpha = \beta \\ & ||C_i(B\alpha + \mu)||_2^2 = L_i, i = 1, 2, \ldots, m \end{aligned} \tag{4}$$

其增强拉格朗日方程为

$$\begin{aligned} L_1(\alpha, \beta, \gamma, \lambda_1, \lambda_2, \eta) = & ||\gamma||_1 + \theta||\beta||_1 + \\ & \lambda_1^T[\gamma - x + M(B\alpha + \mu)] + \lambda_2^T(\alpha - \beta) + \\ & \frac{\eta}{2}[||\gamma - x + M(B\alpha + \mu)||^2 + ||\alpha - \beta||^2] \end{aligned}$$

原优化问题可以等价为

$$\begin{aligned} & \min_{\alpha,\beta,\gamma} L_1(\alpha, \beta, \gamma, \lambda_1, \lambda_2, \eta) \\ & s.t. \quad ||C_i(B\alpha + \mu)||_2^2 = L_i, i = 1, 2, \ldots, m \end{aligned}$$

γ与β的求解

舍弃拉格朗日增强方程中与γ,β无关的变量后

$$\gamma^{k+1} = \underset{\gamma}{argmin}\, ||\gamma||_1 + \frac{\eta_k}{2}||\gamma - [x - M(B\alpha^k + \mu) - \frac{\lambda_1^k}{\eta_k}]||^2$$

$$\beta^{k+1} = \underset{\beta}{argmin}\, ||\beta||_1 + \frac{\eta_k}{2\theta}||\beta - (\frac{\lambda_2^k}{\eta_k} + \alpha^k)||^2$$

是无约束优化问题,可以符合DCP规则采用CVXPY直接求解,但以下有更简单的方法

定理

对于

$$prox_{f_i,\sigma}(w) = \underset{x_i}{argmin}\, f_i(x_i) + \frac{\sigma}{2}||x_i - w||^2$$

当f_i为1-范数时

$$prox_{f_i,\sigma}(w) = sgn(w)max(|w| - \sigma^{-1}) \tag{5}$$

γ^{k+1}, β^{k+1}带入后可以得到闭合解形式,运算速度更快.

α的求解

舍弃拉格朗日增强方程中与α无关的变量后

$$\alpha^{k+1} = \underset{\alpha}{argmin} \quad z^T W z$$
$$s.t. \quad z^T \Omega_i z = 0, i = 1, \ldots, m$$

$$z = \begin{pmatrix} \alpha^T & 1 \end{pmatrix}^T$$

$$W = \begin{pmatrix} B^T M^T M B + I & 0 \\ 2[(\gamma^{k+1} - x + M\mu + \frac{\lambda_1^k}{n_k})^T M B - \beta^{k+1} + \frac{\lambda_2^k}{\eta_k}] & 0 \end{pmatrix}$$

$$\Omega_i = \begin{pmatrix} B^T C_i^T C_i B & B^T C_i^T C_i \mu \\ \mu^T C_i^T C_i B & \mu^T C_i^T C_i \mu - L_i \end{pmatrix}$$

由$x^T A y = tr(x^T A y) = tr(A y x^T)$,可以设$Q = zz^T$,则

$$\min_{Q} \quad tr(WQ)$$
$$s.t. \quad tr(\Omega_i Q) = 0, i = 1, \ldots, m$$
$$Q \geq 0, rank(Q) \leq 1$$

α的求解

为借用ADMM方法,引入P

$$\begin{aligned} \min_{Q,P} \quad & tr(WQ) \\ s.t. \quad & tr(\Omega_i Q) = 0, i = 1, \ldots, m \\ & P = Q, rank(P) \leq 1, P \geq 0 \end{aligned}$$

其增强拉格朗日方程为

$$L_2(Q, P, G, \delta) = tr(WQ) + tr(G^T(Q-P)) + \frac{\delta}{2}||Q-P||_F^2$$

- Q^{l+1}**求解**

$$\begin{aligned} Q^{l+1} = & argmin \;\; L_2(Q, P^l, G^l, \delta_l) \\ & tr(\Omega_i Q) = 0, i = 1, \ldots, m \end{aligned}$$

  符合DCP规则,故直接用CVXPY求解决(具体参考补充内容)
- P^{l+1}**求解**,参考补充内容,最终可以化为闭合解形式

$$\begin{aligned} P^{l+1} = & argmin \, ||P - (Q^{l+1} + \frac{2}{\delta_l} G^l)|| \\ & P \geq 0 \\ & rank(P) \leq 1 \end{aligned}$$

相机参数求解

同上,为采用ADMM方法,引入R,改写原问题为

$$\min_{R,m_1,m_2} \quad ||R||_1$$
$$s.t. \quad R = X - \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} Y,\ m_1^T m_2 = 0$$

增强拉格朗日方程为

$$
\begin{aligned}
& L_3(R, m_1, m_2, H, \zeta, \tau) \\
& = ||R||_1 + tr\left(H^T \left[\begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} Y + R - X\right]\right) + \zeta(m_1^T m_2) \\
& + \frac{\tau}{2}\left[\left|\left|\begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} Y + R - X\right|\right|_F^2 + (m_1^T m2)^2\right]
\end{aligned}
$$

R和m_1, m_2求解

- **R求解**

$$R^{k+1} = \underset{R}{argmin} \, ||R||_1 + \frac{\tau_k}{2} \left|\left| R + \begin{pmatrix} (m_1^k)^T \\ (m_2^k)^T \end{pmatrix} Y - X + \frac{H^k}{\tau_k} \right|\right|_F^2$$

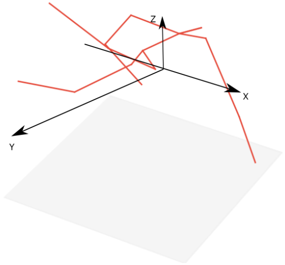
同样,参考公式(5),有闭合解

- m_1**求解**

$$m_1^{k+1} = \underset{m_1}{argmin} \left|\left| \begin{pmatrix} m_1^T \\ (m_2^k)^T \end{pmatrix} Y + R^{k+1} - X + \frac{H^k}{\tau_k} \right|\right|_F^2 + (m_1^T m_2^k + \frac{\zeta^k}{\tau_k})^2$$

符合DCP规则,可以用CVXPY求解,其次通过观察可知目标函数是个平滑的凸函数,故通过令一阶导数为零可以得到闭合解.m_2同理

数据标准化



3D数据标准化, 对于世界坐标系中的y,先将其变换到以物体为基准的坐标系,然后标准化

- 以左肩到右肩为x轴,
- 以两肩中点到腰为y轴
- z轴是x与y的叉乘.
- 然后以各个关节点各轴上点取平均得到$\mu \in R^3$,以其为原点.
- 对于每个关节点,求x_i, y_i, z_i坐标的平方和$s_i = x_i^2 + y_i^2 + z_i^2$,令$S = \sqrt{\sum_i s_i}$,$S$作为归一化系数,对于每个关节点,坐标变换为$x_i/S, y_i/S, z_i/S$

2D数据标准化,

- 然后以各个关节点各轴上点取平均得到$\mu \in R^2$,以其为原点.
- 对于每个关节点,求x_i, y_i坐标的平方和$s_i = x_i^2 + y_i^2$,令$S = \sqrt{\sum_i s_i}$,S作为归一化系数,对于每个关节点,坐标变换为$x_i/S, y_i/S$

字典学习

字典学习采用spams库提供算法,算出B,注意其中3D坐标已经按上述归一化,得到字典矩阵B为45×200

```
import spams
param = { 'K' : 200, 'lambda1' : 0.01, 'iter' :300}
B=spams.trainDL(np.asfortranarray(poses3Ds),**param)
```

由2D关节点信息推断3D姿态

1. **获得骨骼长度** 实现中,首先要确定各个关节点间距离,即骨骼长度.可人工设置,实验中直接由3D姿态坐标计算出.
2. **获得初始**α, 字典学习中得到的字典矩阵B为45×200,42(3×15)是关节点坐标,所以α是200×1的稀疏向量.迭代过程中需要初始化,如果以200维数据开始优化,效率低下.所以先用快速求解lasso问题的算法得出大概的α,再在此基础上开始优化求解.

```
a = spams.lasso(np.asfortranarray(x),np.asfortranarray(M@B),
     return_reg_path = False,lambda1=0.01)
```

3. α求解过程,引入Q, P与上文描述一致,γ, β亦于上文一致

相机参数求解

实践中发现利用迭代优化方法求解相机参数效率低下,且在实践中发现会有不收敛的现象,以下采用一种有闭合解的近似方法.加速了算法,在实践中相比于迭代优化的方法表现更好.

$$P = \begin{pmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{R}' & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{pmatrix}$$

$$= \begin{pmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{r}_1^T & t_1 \\ \boldsymbol{r}_2^T & t_2 \\ \boldsymbol{0}^T & 1 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} s_1 \boldsymbol{r}_1^T \\ s_1 \boldsymbol{r}_2^T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} s_1 t_1 \\ s_2 t_2 \end{pmatrix}$$

$$= \begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{r}_1^T \\ \boldsymbol{r}_2^T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_1' \\ t_2' \end{pmatrix}$$

因为3D坐标和2D坐标都是以均值点作为原点,所以t_1, t_2为0

$$\begin{pmatrix} x \\ y \end{pmatrix} = \boldsymbol{s} \begin{pmatrix} \boldsymbol{r}_1^T \\ \boldsymbol{r}_2^T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \boldsymbol{s}\boldsymbol{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \boldsymbol{M} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

相机参数求解

假设$x \in R^2$,$y \in R^3$,理想情况下

$$\boldsymbol{M}y = x$$

近似估计$\boldsymbol{M}$为

$$\boldsymbol{M'} = xy^T(yy^T)^{-1}$$

一般条件下,这样得出的矩阵不满足M是正交矩阵的条件,且$\boldsymbol{M'}y \neq x$, 对$\boldsymbol{M}$求RQ分解

$$\boldsymbol{M'} = \begin{pmatrix} 0 & s'_1 & \eta \\ 0 & 0 & s'_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{r'^T_3} \\ \boldsymbol{r'^T_1} \\ \boldsymbol{r'^T_2} \end{pmatrix}$$

实验中η相对s_1, s_2较小,且接近于0,所以令$\eta = 0$(实践中令$\eta = 0.05\eta$效果较好),再反向求M得到估计值

$$\boldsymbol{M'} = \begin{pmatrix} 0 & s'_1 & 0 \\ 0 & 0 & s'_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{r'^T_3} \\ \boldsymbol{r'^T_1} \\ \boldsymbol{r'^T_2} \end{pmatrix}$$

$$= \begin{pmatrix} s'_1 & 0 \\ 0 & s'_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{r'^T_1} \\ \boldsymbol{r'^T_2} \end{pmatrix}$$

优缺点

优点:

- 相对于深度学习,其学习速度快.可以用于小样本的情况
- 学习过程中,不需要2D图像和3D姿态信息对,只需要3D姿态信息,数据相对容易获取.

缺点:

- 许多参数需要细致设置.
- 推断速度慢,迭代求解,可并行化程度低.
- 需要人工设置的各个关节点间距离,即骨骼长度.