

1 Appendix

1.1 SMPL(pkl file)

- **SMPLModel(path)**: 24 joints model （需要从 pkl 文件中读取）。

Attributes	Explanation	Type
pose	θ	np.array(24, 3)
beta	β	np.array(10)
trans	世界坐标模型平移量	np.array(3)
verts	模型所有节点的坐标	np.array(6890, 3)
parent	joints 的父节点	dict{int: int}
J_regressor	模型 joints 的 Regressor	
J	joints 的世界坐标	np.array(24, 3)
R	pose θ 的 Rodrigues 旋转矩阵	np.array(3, 3)
set_params(pose, beta, trans)	重新设置参数信息	<i>function</i>
compute_R_G()	根据 θ 计算所有关节的 G 旋转矩阵	<i>function</i> \rightarrow np.array(24, 4, 4)
do_skinning(G)	根据 G 对全体顶点作用 pose	<i>function</i>
update()	计算 G，作用 G	<i>function</i>
rodrigues(r)	将旋转向量 r 转化为 Rodrigues 旋转矩阵	<i>function</i> \rightarrow np.array(3, 3)

1.2 amc_parser

Attributes	Explanation	Type
parse_asf(path)	解析 asf 文件得到整个关节模型 (joints)	<i>function</i> \rightarrow dict{name: joint}
parse_amc(path)	解析 amc 文件得到关节点的逐帧旋转量 (frames)	<i>function</i> \rightarrow list[dict{name: list}]

1.3 skeleton

- **Joint(name, direction, length, axis, dof, limits)**: 将 asf 中的骨骼模型转化为 asf 关节模型。

Attributes	Explanation	Type
name	关节名称	string
direction	T-pose 状态下骨骼方向的正则化向量	np.array(3)
length	骨骼长度	scalar
axis	初始状态下骨骼在世界坐标下的旋转向量	list[float]
C	axis 的 Rodrigues 旋转矩阵形式	np.array(3, 3)
Cinv	C 的逆矩阵	np.array(3, 3)
parent	父节点	Joint
children	子节点	list[Joint]
coordinate	关节点的世界坐标	np.array(3)
matrix	世界坐标下关节点的旋转信息总和（父节点+此节点）	np.matrix(3, 3)
relative_R	动作变换 $C \times (\text{motion rotation}) \times C_{\text{inv}}$	np.array(3, 3)
set_motion(motion)	根据 motion 计算所有节点的 coordinate 和 matrix	<i>function</i>
to_dict()	得到整个模型的信息	<i>function</i> \rightarrow dict{name: Joint}
reset_pose()	重置模型得到 T-pose 状态（relative_R = I）	<i>function</i>

- **SMPLJoints(idx)**: SMPL 模型的关节点对象。

Attributes	Explanation	Type
idx	24 joints SMPL 模型的关节编号	int
to_parent	父节点到此节点的向量	np.array(3, 3)
parent	父节点	SMPLJoints
coordinate	关节点的世界坐标	np.array(3)
children	子节点	list[SMPLJoints]
align_R	SMPL T-pose 到 asf 模型的纠正量	np.array(3, 3)
motion_R	amc 文件中该节点的旋转量	np.array(3, 3)
init_bone()	计算初始状态的 to_parent	<i>function</i>
set_motion_R(motion)	根据 motion 计算所有节点的旋转矩阵	<i>function</i>
update_coord()	更新所有节点的 coordinate	<i>function</i>
to_dict()	得到整个模型的信息	<i>function</i> \rightarrow dict{idx: SMPLJoints}
export_G()	根据 motion_R 和 align_R 计算旋转矩阵 G	<i>function</i> \rightarrow np.array(3, 3)

1.4 imitator

- **imitator(asf_joints, smpl)**: 将 asf/amc 数据集迁移到 SMPL 模型。

Attributes	Explanation	Type
asf_joints	asf 文件解析得到的关节点信息 (joints)	dict{name: joint}
smpl	pkl 文件解析得到的 SMPL 模型	SMPLModel
setup_smpl_joints()	初始化 SMPL 模型节点 (joints)	<i>function</i> \rightarrow dict{idx: SMPLJoints}
align_smpl_asf()	校准 asf 模型到 SMPL （align_R）	<i>function</i>
compute_rodrigues(x, y)	计算 align_R 满足 $y = Rx$	<i>function</i> \rightarrow np.array(3, 3)
imitate(motion, translate=False)	根据 amc motion 进行模型演示	<i>function</i>
set_asf_motion(motion, translate)	同上	<i>function</i>
asf_to_smpl_joints(translation)	将 asf 模型的 pose 迁移到 SMPL 模型	<i>function</i>
map_R_asf_smpl()	将 asf 模型的旋转量转移到 SMPL 模型	<i>function</i> \rightarrow (dict{idx: R}, T)