Motion Keyframe Interpolation for Any Human Skeleton via Temporally Consistent Point Cloud Sampling and Reconstruction

Supplementary Material

1 Algorithm Details

For all functions listed described below, the temporal axis, i.e. t, is not relevant and can be arbitrary.

1.1 Forwards Kinematics

A 3D point can be rotated around the origin, i.e. $[0, 0, 0]$, using the quaternion rotation function $QR(\mathbf{p}, \mathbf{q}) = \mathbf{qpq}^{-1}$, where quaternion $\mathbf{q}$ assumes the standard quaternion form $\mathbf{q}_w + \mathbf{q}_x\mathbf{i} + \mathbf{q}_y\mathbf{j} + \mathbf{q}_z\mathbf{k} = 0$, and the xyz coordinates $[\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z]$ of 3D point $\mathbf{p}$ respectively take the form $\mathbf{p}_x\mathbf{i} + \mathbf{p}_y\mathbf{j} + \mathbf{p}_z\mathbf{k} = 0$. The function can be computed using any variety of quaternion multiplication, i.e. *Hamilton product*, solvers [1].

We can retrieve the set of global joint positions for a given skeleton S iteratively using the FK algorithm. That is,

Algorithm 2 Forward kinematics of full skeletal hierarchy

```
1: Inputs:
2: $\mathcal{B} = \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, ...\}$; Joint set of $S$, where $\mathbf{v}_n$ is parented to $\mathbf{v}_m$. Assume $m < n$.
3: $\mathcal{Q} = \{\mathbf{q}_n | \mathbf{v}_n \in \mathcal{B}\}$; Quaternion rotations for each joint
4: $\mathbf{p}_0$; Root joint 3D position
5: Outputs:
6: $\{\mathbf{p}_n \mid \mathbf{v}_n \in \mathcal{B}\}$; Global 3D positions for each joint
7:
8: for $\mathbf{v}_n \in \mathcal{B} \setminus \mathbf{v}_0$ do
9:     $\mathbf{p}_n \leftarrow \mathbf{p}_m + QR(\mathbf{H}_n, \mathbf{q}_m)$
10: end for
```

1.2 Rest Pose Augmentation

We apply Algorithm 1 from the main paper, shortened as $IK(\mathbf{p}_{\text{IK}}, \mathbf{T}) \rightarrow \mathbf{q}$, to produce rest pose augmentations. Likewise, we shorten Algorithm 2 as $FK(\mathcal{B}, \mathcal{Q}, \mathbf{p}_0) \rightarrow \{\mathbf{p}_n \mid \mathbf{v}_n \in \mathcal{B}\}$. The rest pose augmentation method only augments quaternion rotations to be used on the original skeletal offsets, as shown in Algorithm 3. This algorithm does not modify the structure of the skeleton used in training.

Algorithm 3 Rest pose augmentation with 3D&$\mathbb{C}$

```
1: Inputs:
2: $\mathcal{B} = \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, ...\}$; Joint set of $S$, where $\mathbf{v}_n$ is parented to $\mathbf{v}_m$. Assume $m < n$.
3: $LR$ = Set of all symmetrical joint pairs $\mathbf{v}_L$ and $\mathbf{v}_R$. Assume the symmetry to occur
   on the $x$-axis.
4: $\mathcal{Q} = \{\mathbf{q}_n \mid \mathbf{v}_n \in \mathcal{B}\}$; Quaternion rotations for each joint
5: $\Gamma$ = Tail augmentation sampling range
6: Outputs:
7: $\{\hat{\mathbf{q}}_n \mid \mathbf{v}_n \in \mathcal{B}\}$: Augmented quaternion rotations
8:
9: $\mathcal{P} \leftarrow FK(\mathcal{B}, \mathcal{Q}, [0,0,0])$          ▷ Root-local FK positions $\{\mathbf{p}_n \mid \mathbf{v}_n \in \mathcal{B}\}$
10: $\mathcal{O} \leftarrow \{\Delta_n \leftarrow [0,0,0] \mid \mathbf{v}_n \in \mathcal{B}\}$          ▷ Set of rest pose augmented offsets
11:
12: for $\mathbf{v}_L, \mathbf{v}_R \in LR$ do
13:     $\Delta_L \leftarrow \mathcal{U}_{[-\Gamma,\Gamma]} \in \mathbb{R}^3$          ▷ Uniform sampling of augmentation range
14:     $\Delta_R \leftarrow \Delta_L \odot [-1,1,1]$          ▷ Symmetrical sampling
15: end for
16:
17: $\mathcal{E} \leftarrow \{\mathbf{e}_n \leftarrow QR(\mathbf{T}_n, \mathbf{q}_n) + \mathbf{p}_n \mid \mathbf{v}_n \in \mathcal{B}\}$          ▷ Root- local 3D joint tail positions
18: $\hat{\mathbf{T}} \leftarrow \{\hat{\mathbf{T}}_n \leftarrow \mathbf{T}_n + \Delta_n \mid \mathbf{v}_n \in \mathcal{B}\}$
19:
20: for $\mathbf{v}_n \in \mathcal{B}$ do
21:     if $n = 0$ then
22:         $\hat{\mathbf{p}}_n \leftarrow [0,0,0]$
23:     else
24:         $\hat{\mathbf{p}}_n \leftarrow QR(\mathbf{T}_n, \hat{\mathbf{q}}_m) + \hat{\mathbf{p}}_m$
25:     end if
26:     $\mathbf{q}_\Delta \leftarrow IK(\mathbf{e}_n - \mathbf{p}_n, \mathbf{e}_n - \hat{\mathbf{p}}_n, \mathbf{i}) \times \mathbf{q}_n$          ▷ $\mathbf{i}$ = imaginary unit
27:     $\hat{\mathbf{q}}_n \leftarrow IK(\mathbf{e}_n - \hat{\mathbf{p}}_n, FK(\hat{\mathbf{T}}_n, \mathbf{q}_\Delta), \mathbf{i}) \times \mathbf{q}_\Delta$
```

1.3 Video Demonstrations

We provide additional demonstrations of our method in the attached video. Included are animated visualisations of our point cloud sampling strategy, its σ parameter, the workings of our KNN-Loss function, and motion interpolation examples with our indirect point cloud-based supervision.

2 Supplementary Experimental Results

2.1 Additional Motion Categories for Motion Interpolation

Below, we test additional motion categories for the motion interpolation experiment described from Table 1. These results reinforce our observations as denoted in Section 4.3.

Motion category	**Method**	**L2P↓**			**L2Q↓**			**NPSS↓**		
		5	15	30	5	15	30	5	15	30
Acrobatics	LERP	0.0634	0.3299	0.7154	0.1387	0.4702	0.8936	0.1526	0.7192	1.9218
	$\text{TG}_{\text{complete}}$	0.0956	0.4297	0.7530	0.1780	0.5625	0.9027	0.1587	0.7344	1.8669
	BERT	0.0720	0.3573	0.6992	0.1419	0.4945	0.8632	0.1677	0.6980	1.9243
	CITL	0.0593	0.2635	0.5252	0.1381	0.4150	0.7279	0.1463	0.5893	1.3274
	PC-MRL	0.0685	0.3249	0.6978	0.1480	0.4748	0.8869	0.1723	0.7226	1.8923
Construction	LERP	0.0118	0.0575	0.1363	0.0318	0.1210	0.2382	0.0214	0.1011	0.2648
	$\text{TG}_{\text{complete}}$	0.0221	0.0854	0.1850	0.0580	0.1927	0.3005	0.0376	0.1191	0.2763
	BERT	0.0201	0.0837	0.1722	0.0503	0.1865	0.2980	0.0458	0.1232	0.2935
	CITL	0.0162	0.0731	0.1666	0.0375	0.1405	0.2758	0.0252	0.1143	0.2874
	PC-MRL	0.0198	0.0558	0.1342	0.0417	0.1271	0.2408	0.0373	0.1221	0.2864
Push	LERP	0.0347	0.1679	0.3723	0.0803	0.2930	0.5272	0.0753	0.2837	0.7367
	$\text{TG}_{\text{complete}}$	0.0360	0.1722	0.3521	0.0853	0.2965	0.5064	0.0779	0.3014	0.6355
	BERT	0.0357	0.1656	0.3478	0.0831	0.2894	0.4932	0.0760	0.2813	0.6241
	CITL	0.0331	0.1418	0.2970	0.0794	0.2637	0.4631	0.0747	0.2651	0.5827
	PC-MRL	0.0341	0.1539	0.3307	0.0799	0.2816	0.4846	0.0785	0.2973	0.6995
Salsa	LERP	0.0545	0.3044	0.6945	0.1263	0.5186	1.0707	0.1301	0.7475	2.5910
	$\text{TG}_{\text{complete}}$	0.0639	0.3106	0.6565	0.1730	0.5179	1.0671	0.1689	0.7398	2.4116
	BERT	0.0604	0.3006	0.6550	0.1599	0.5020	1.0507	0.1542	0.7575	2.4587
	CITL	0.0515	0.2735	0.6388	0.1312	0.4994	1.0254	0.1478	0.6771	2.1076
	PC-MRL	0.0539	0.2861	0.6340	0.1358	0.5100	1.0206	0.1545	0.7173	2.3245

2.2 Training Objective Configurations

We conduct an ablation study on the effect of different K values in our KNN-Loss objective, as well as end-effector loss. All experiments are conducted with $K = 8$ and end-effector loss enabled unless stated otherwise:

Method	**L2P↓**			**L2Q↓**			**NPSS↓**		
	5	15	30	5	15	30	5	15	30
No end-effector	0.0760	0.3438	0.6730	0.1589	0.5487	0.9093	0.1400	0.5621	1.3016
$K = 2$	0.0825	0.3664	0.6901	0.1549	0.5522	0.9020	0.1495	0.6098	1.4250
$K = 4$	0.0807	0.3617	0.6814	0.1559	0.5577	0.9168	0.1487	0.5808	1.3573
Our configuration	0.0752	0.3443	0.6662	0.1561	0.5495	0.9088	0.1397	0.5572	1.3015

References

1. Hamilton, W.R.: Lxxviii. on quaternions; or on a new system of imaginaries in algebra: To the editors of the philosophical magazine and journal. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **25**(169), 489–495 (1844) 1