GaitCloud: Leveraging Spatial-temporal Information for LiDAR-base Gait Recognition with A True-3D Gait Representation

Supplementary Material

Shaoxiong Zhang, Hiromitsu Awano, Takashi Sato

Kyoto University

zhang.shaoxiong.86c@st.kyoto-u.ac.jp

The source code of this work is available at https://github.com/seagrgz/GaitCloud-master.git

1. Gait Examples in SUSTech1K

1.1 Demonstration of Gait Diversity

In this section, we provide a comprehensive demonstration of samples with all attributes in SUSTech1K, including raw RGB images, silhouettes, depth images, voxelized frames, and GaitCloud. The names of distributes are specified as follows: Gallery---Normal sample for referring in inference of variance experiments, nm---Normal samples for probe set in inference, bg---Bag, cl---Clothing, cr---Carrying, ub---Umbrella, uf---Uniform, oc---Occlusion, nt---Night. Both voxelized frames and GaitCloud are rotated to face the same direction, following the proposed gait rotation workflow. GaitCloud is generated using the same voxelization and temporal integration processes described in the paper.

Variance. Figure 1 presents examples of gait samples categorized by walking conditions (variance). Samples with the same attributes exhibit detailed variations, indicating they are not strictly constrained

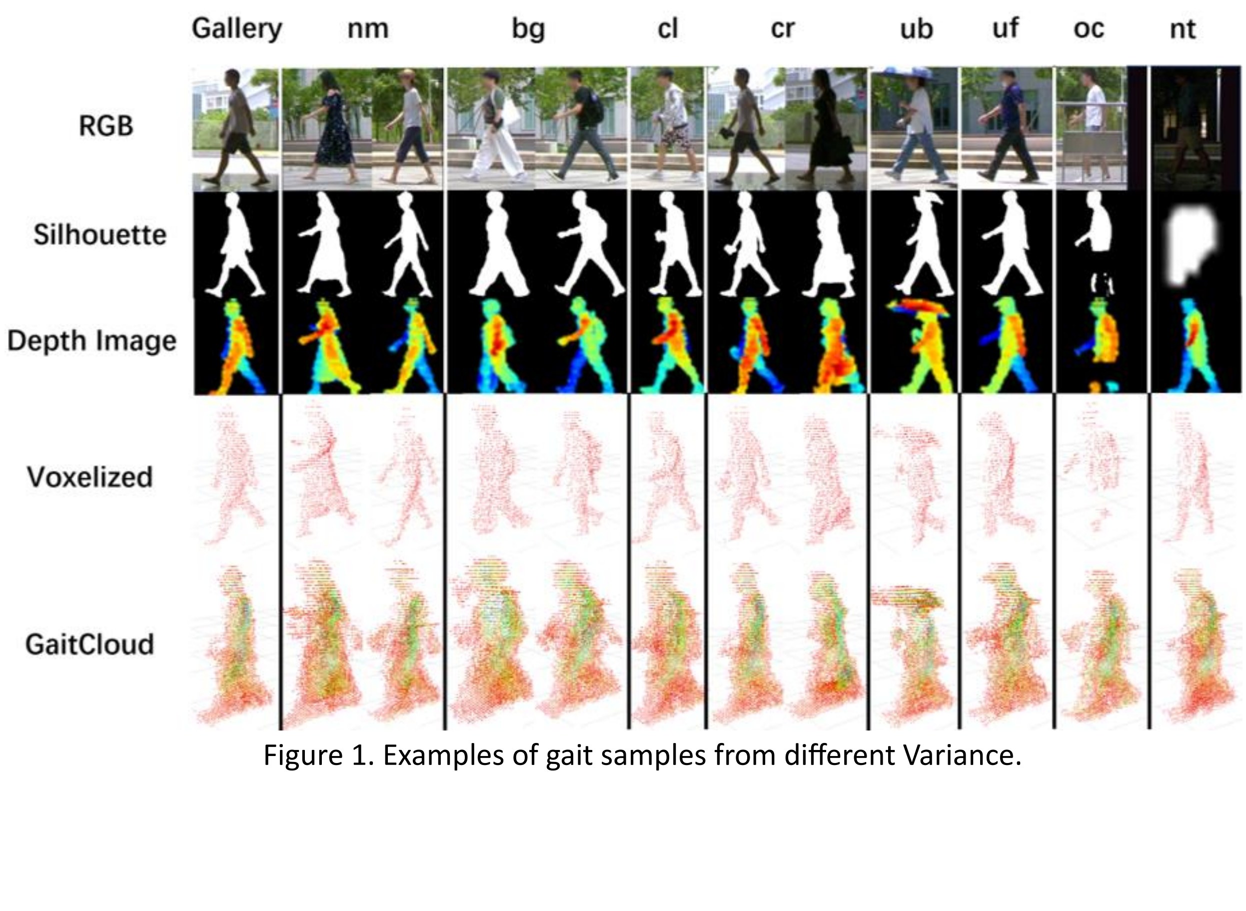


Figure 1. Examples of gait samples from different Variance.

by predefined criteria. Some samples also have multiple attributes such as “01-bg-ub”, meaning the subject is carrying both a bag and an umbrella, and will be considered in the inference for both the Bag and Umbrella attributes.

View. Figure 2 presents samples from different views of the same identity, all within the 00-nm attribute. Both voxelized frames and GaitCloud representations are generated using the same process as in Variance.

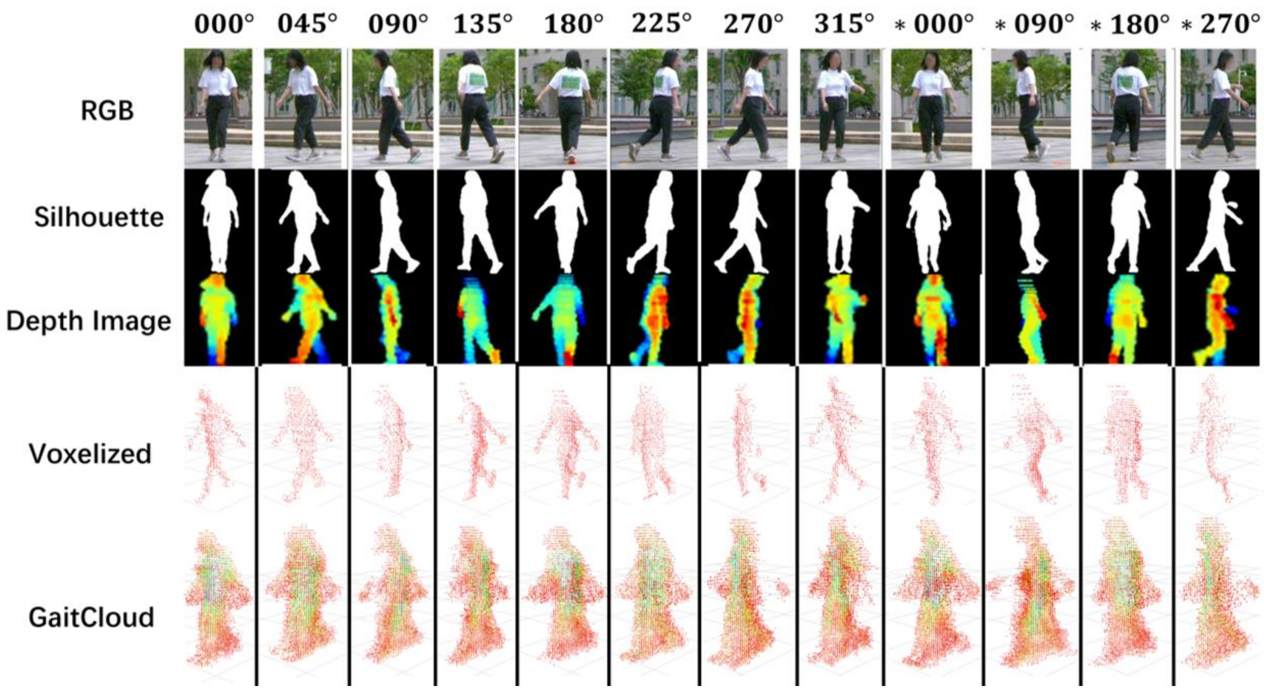
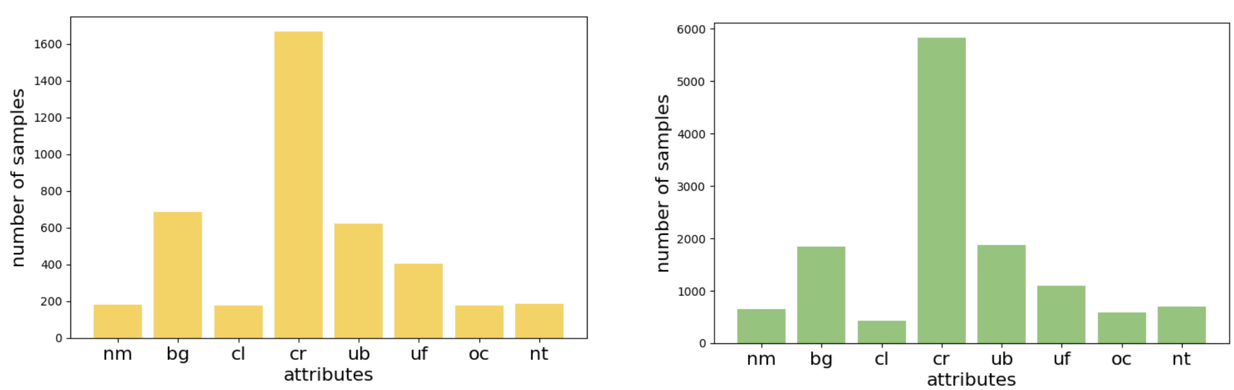


Figure 2. Examples of gait samples from different Views.

1.2 Attribute Distribution

Figure 3 shows the number of samples in each attribute in the dataset. Similar distribution imbalances can be observed both in training and test sets. These imbalances may lead the model to prioritize adapting to attributes with larger sample populations and weaken the impact of a small number of samples on the overall accuracy of the inference.



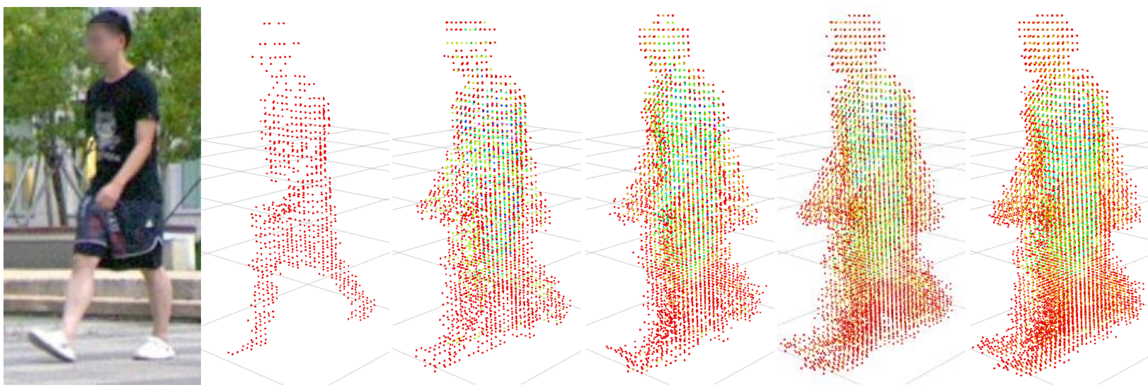
(a) Number of samples in each attribute of training (b) Number of samples in each attribute of test

Figure 3 Data distributions on (a) training set and (b) test set.

1.3 GaitCloud Representations with Varying Numbers of Frames.

Figure 4 presents GaitClouds created with different numbers of frames used in the ablation study on frame numbers. As the number of frames increases, the complexity of the point cloud contours also increases, capturing more gait-related statistical features for high-performance recognition.

We only investigate the largest frame number of 30 since the length of most sequences in SUSTech1K is around 25. Selecting a number > 40 for sample frames may lead to an overabundance of self-replicating samples according to the random temporal cropping procedure, rendering the test results meaningless.



RGB **1 frame** **5 frames** **10 frames** **20 frames** **30 frames**

Figure 4 GaitCloud created with different number of frames

2. Supplementary Results

2.1 Computational Efficiency

Table 1 shows the data size of a GaitCloud and a depth image sequence used for experiments. The data size of a depth image sequence scales with the number of frames used in a sample, whereas the data size of GaitCloud remains constant regardless of the number of frames. This property allows GaitCloud to achieve high recognition performance without increasing computational complexity.

Table 2 shows the computational comparison results of different models. All training ran on the

	GaitCloud	Depth Images	GaitCloud(temporal)
Shape	$40 \times 40 \times 60$	$10 \times 3 \times 64 \times 64$	$10 \times 64 \times 40 \times 40$
Type	Uint8	Uint8	Uint8
Size	101KB	121KB	1004KB

Table 1 Comparison on different gait representations

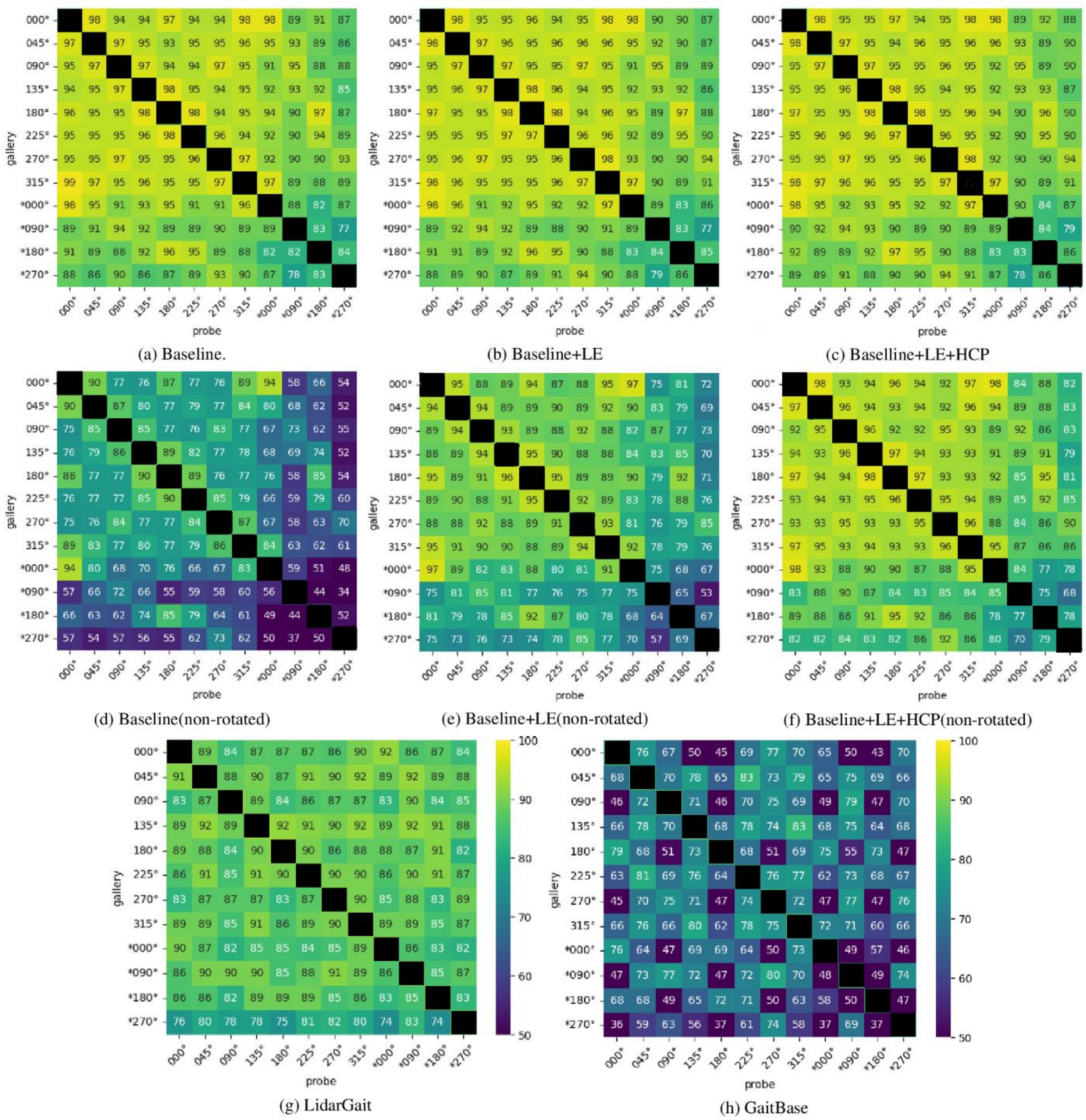
	GaitCloud	LidarGait	3D LidarGait
Parameters	43M	8M	17M
GPU mem.(MB)	2864	1356	13534
Forward time(s)	0.0171	0.004	0.1209

Table 2 Computational comparison

same GPU server with a single RTX 4090. Most of the additional parameters in GaitCloud come from HCP. Despite the large number of parameters, the computational overhead of GaitCloud is trivial, since the kernel size of HCP is equal to input features and the exact operation is equivalent to the summation with learnable weight.

2.2 Detailed Results on Cross-view Experiments

We demonstrate results from cross-view experiments for all experimental groups presented in the main results with detailed heatmaps, shown in Figure 4. Baseline+LE+HCP achieves the best cross-view accuracy over all groups.



(a) Baseline. (b) Baseline+LE (c) Baselline+LE+HCP

(d) Baseline(non-rotated) (e) Baseline+LE(non-rotated) (f) Baseline+LE+HCP(non-rotated)

(g) LidarGait (h) GaitBase

Figure 5 Detailed cross-view results from each group.

2.3 Detailed Results on Layer Reduction Experiments.

Table 3 shows the comparative results of models with and without the modules. 'Layers' indicates the number of residual blocks to construct the feature encoder. The Detailed results of cross-view experiments are shown in Figure 6.

Model	Layers	Probe Attributes (Rank-1 accuracy)								Overall	Views
		Normal	Bag	Clothing	Carrying	Umbrella	Uniform	Occlusion	Night	$Rank1$	$Rank1$
baseline	2	88.91	91.55	82.41	92.02	90.76	92.92	94.05	92.06	91.40	93.32
	1	84.65	89.06	81.48	89.21	88.09	89.56	91.50	88.31	88.61	87.5
baseline +LE+HCP	2	87.23	91.50	79.63	90.84	89.37	91.20	93.37	91.20	90.25	93.35
	1	86.02	89.87	80.32	89.33	88.89	89.20	91.16	89.03	88.87	92.45

Table 3 Accuracy on baseline and proposed models with different number of encoder blocks.

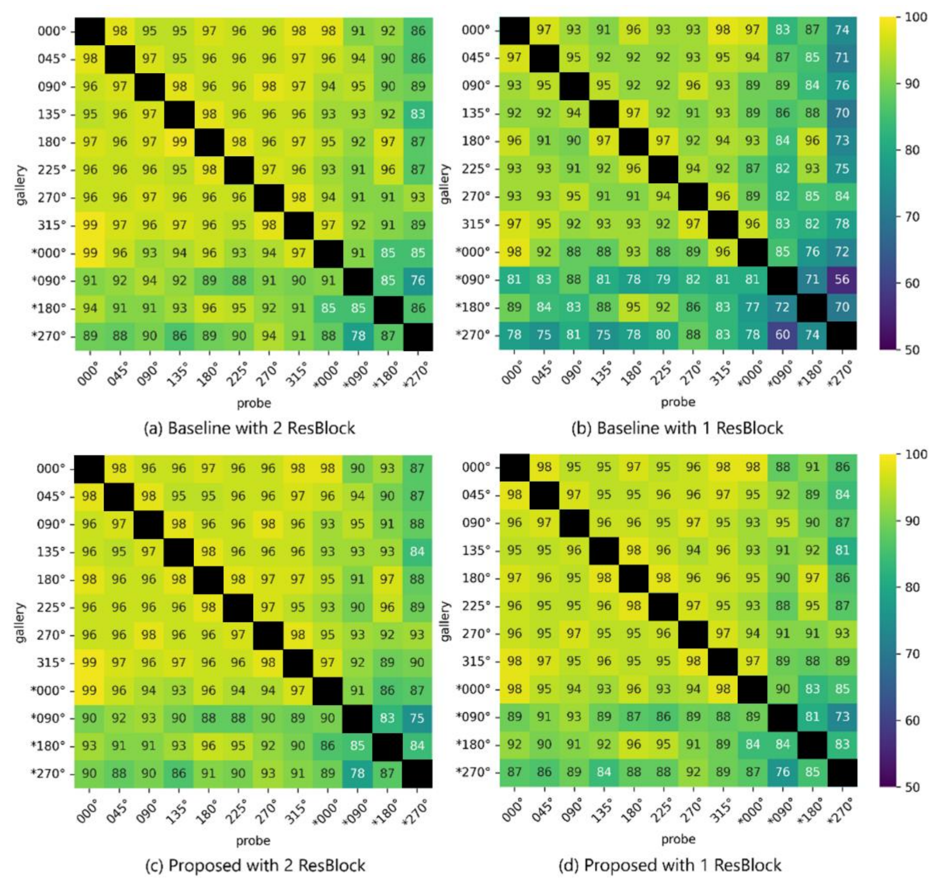


Figure 6 Detailed results in cross-view experiments