GarmentCodeData (v2): A Dataset of 3D Made-to-Measure Garments With Sewing Patterns

Dataset

Author(s):
Korosteleva, Maria; Kemper, Fabian; Wenninger, Stephan; Koller, Jasmin; Zhang, Yuhan; Botsch, Mario; Sorkine-Hornung, Olga

Publication date:
2024-09-04

Permanent link:
https://doi.org/10.3929/ethz-b-000690432

Funding acknowledgement:
101003104 - Sustainable Algorithmic Modeling of Personalized Garments (EC)

GarmentCodeData (v2): dataset documentation

Maria Korosteleva, Fabian Kemper, Stephan Wenninger, Jasmin Koller, Yuhan Zhang, Mario Botsch, Olga Sorkine-Hornung

Overview

The dataset is shared on libdrive and includes the following items:

- `./5000_body_shapes_and_measures.tar.gz` – the body shapes dataset
- `./GarmentCodeData` – the main garment dataset
- `GarmentCodeData_official_train_valid_test_data_split.json` – official data split of the garment dataset for promoting comparable training results. It only includes the designs that are present for both neutral and random body shapes.
- `./neutral_body` – the neutral body model and corresponding measurements.

Citing

If you are using our dataset in your work, please, cite our paper:

```
@inproceedings{GarmentCodeData2024,
  author = {Korosteleva, Maria and Kesdogan, Timur Levent and Kemper, Fabian and Wenninger, Stephan
and Koller, Jasmin and Zhang, Yuhan and Botsch, Mario and Sorkine-Hornung, Olga},
  title = {{GarmentCodeData}: A Dataset of 3{D} Made-to-Measure Garments With Sewing Patterns},
  booktitle={Computer Vision -- ECCV 2024},
  year = {2024},
  keywords = {sewing patterns, garment reconstruction, dataset}
}
```

Changes in v2

The data was re-generated following the improvements in the generation pipeline and feedback from early users. The dataset is based on the same sample set of design parameters, however, the set of failed and successful data points after simulation varies slightly from GarmentCodeData (v1).

- Improvements in the stitching process now lead to fewer fails due to incorrect stitch matching.
- New checks for degenerate triangles filter the datapoints with extremely thin triangles. As a result, the overall geometry quality is improved.
- Additional effort was put into ensuring the matching of the side stitches on full-body garments (jumpsuits, dresses, w/ and w/o waistbands) for improved sewing pattern designs.

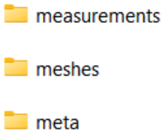
Body shapes dataset

Separately, we share a selection of body shape samples used during garment data generation and the neutral (average) body shape. The dataset contains 5000 body shapes.

Neutral body shape

Folder structure

- `./measurements/`– contains body measurement files for each body sample (`.yaml`)
- `./meshes/` – 3D models for body samples (`.obj`).
    - Note that each body sample is provided with two mesh representations: "straight" and "apart". The name indicates the leg pose: the "apart" version has its feet placed slightly wider for the convenience of draping pants.
- `./meta/` – contains meta files (`.json`) with the corresponding PCA weights for each body sample



measurements

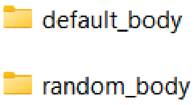
meshes

meta

Garment dataset structure overview

The dataset is grouped into batches of 3200-3500 designs in each (see below for details). Samples across batches are unique and drawn from the same design space, but vary in terms of fabric material used during draping, which is assigned per batch level. Every batch has the same structure:

- `./default_body/` subfolder. Contains sewing patterns fitted to and draped on the neutral body shape
- `./random_body/` subfolder. Contains sewing patterns fitted to and draped on a random body shape (body shapes sampled separately for each design)



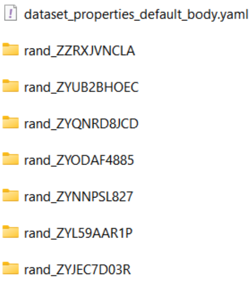
default_body

random_body

Top level

Elements contained in both subfolders correspond to each other: each element in `./default_body/` folder has a counterpart in `./random_body/` subfolder with the same design but different body shape fit. Corresponding elements share the same name.

The division between body types is used on the highest folder level to allow easy parallel processing and data filtering when only the drapes on one body type are needed.



dataset_properties_default_body.yaml

rand_ZZRXJVNCLA

rand_ZYUB2BHOEC

rand_ZYQNRD8JCD

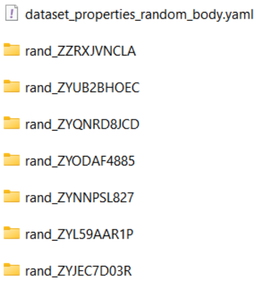
rand_ZYODAF4885

rand_ZYNNPSL827

rand_ZYL59AAR1P

rand_ZYJEC7D03R

./default_body/



dataset_properties_random_body.yaml

rand_ZZRXJVNCLA

rand_ZYUB2BHOEC

rand_ZYQNRD8JCD

rand_ZYODAF4885

rand_ZYNNPSL827

rand_ZYL59AAR1P

rand_ZYJEC7D03R

./random_body/

Each subfolder contains:

- Element folders by name containing all the information related to one drape
- `dataset_properties.yaml` file

Packing strategy

For memory efficiency, the dataset is shared with the contents of each batch's subfolder packed into a `tar.gz` file. The higher-level structure is preserved.

Failed draping cases

In the dataset, we share only the results of the successful drapes to avoid wasting storage and traffic on unusable data. Hence, you may find that `dataset_properties` file contains statistics for the drapes that are not present in the shared dataset.

Datapoints

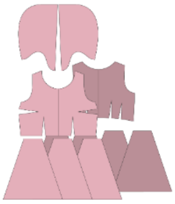
What is a datapoint?

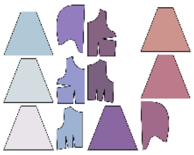
We say that one datapoint in GarmentCodeData corresponds to one design sample and consists of two elements:

- An element in the `./default_body/` subfolder – design fitted to and draped on the neutral body
- An element in `./random_body/` subfolder – the same design fitted to and draped on a random body

Element folder structure

Each element folder contains the following files:

File	Description
{el_name}_design_params.yaml	List of design parameters and their values corresponding to the current design.
{el_name}_body_measurements.yaml	Body measurement values used when creating the given sewing pattern. The file also contains the name of the used body shape sample.
{el_name}_specification.json	Sewing pattern specification in the [Korosteleva and Lee 2021] JSON format.
{el_name}_pattern.svg	Visualization of sewing pattern panels and their relative locations (projected onto 2D) in vector graphics format. 
{el_name}_pattern.png	Rasterized version of the pattern.svg for easy visualization.
{el_name}_boxmesh.ply	3D box mesh (initial state of a garment mesh before draping). The mesh UV's are defined by islands corresponding to sewing pattern panels, which can be used to restore the individual panel meshes.
{el_name}_orig_lens.pickle	List of the garment mesh's edge lengths in the rest state. Used during draping.

{el_name}_sim.ply	Final draped garment mesh. The mesh UV's are defined by islands corresponding to sewing pattern panels, which can be used to identify individual panel meshes in their draped state.	
{el_name}_texture.png	The texture file used for rendering. The fabric grain effect is removed to save storage space but can be added back by overlaying the texture over the fabric grain image provided with the simulation code.	
{el_name}_sim_segmentation.txt	Segmentation of a garment mesh into panels: per-vertex panel labels, with stitch vertices labeled separately. Vertex labels apply to both the box mesh and the draped mesh.	
{el_name}_vertex_labels.yaml	Additional semantic vertex labels, marking vertices that belong to waist stitches, collars or the top border of strapless tops. Mostly needed for attachment constraints. Vertex labels apply to both the box mesh and the draped mesh.	
{el_name}_render_front.png	Render of the final drape with visualized panel segmentation (front view).	
{el_name}_render_back.png	Render of the final drape with visualized panel segmentation (back view).	

Loading UV maps from ply meshes

We use the `.ply` file format for optimizing the storage space. Natively, many geometry editors do not read texture (UV) coordinates from .ply files. However, our `.ply` files do contain texture coordinates. They can be accessed by loading the meshes with `trimesh` library.

```
import trimesh
mesh = trimesh.load_mesh(path_to_mesh, process=False)
tex_image = PIL.Image.open(path_to_texture_img)
tex = trimesh.visual.TextureVisuals(mesh.visual.uv, image=tex_image)
mesh.visual = tex
mesh.show()
```

Loading segmentation and vertex labels for ply meshes

We saved `.ply` meshes using `trimesh` library. It has a known issue of creating vertex duplicates at UV seams. Vertex duplicates break the initial topology and the vertex IDs referenced in segmentation and vertex label files. Luckily, the duplicates are inserted right after the original vertex, so mapping them to the original vertex IDs could be done like this:

```
def v_id_map(vertices):
    v_map = [None] * len(vertices)
    v_map[0] = 0
    for i in range(1, len(vertices)):
        if all(vertices[i - 1] == vertices[i]):
            v_map[i] = v_map[i-1]
        else:
            v_map[i] = v_map[i-1] + 1
    return v_map
```

Assigned fabric materials per batch

During the garment draping step, we use three different fabric materials and assign them per batch; all samples in the same batch use the same material properties. The table below indicates which materials were assigned to which batches:

Material	Batch ids (inclusive)
Smallest bending resistance (b=1.0)	0-6, 21-25
Medium bending resistance (b=100)	7-13, 26-30
Largest bending resistance (b=500)	14-20, 31-35

Dataset properties file

Each batch's subfolders contain the dataset properties file that stores information regarding the data elements in the subfolder. We follow the [Korosteleva and Lee 2021] format for dataset properties. It contains basic information about the batch (name, pattern sample size, names of the used body shape samples dataset and a neutral body 3D model), the flag that protects from re-writing with accidental restarts of a draping step – the 'frozen' flag should be manually set to False to allow restart of the draping. It also contains system information for the machine used during draping.

However, the main purpose of the dataset properties file is to store the configuration and the data statistics for each step of the data generation process. Here is the overview of the contents:

- Generation:
    - Config: randomization seed
    - Statistics: generation time, garment type per drape, panel count per drape
- Simulation:
    - Config: fabric material properties, stopping conditions, enabled simulation features
    - Statistics: list of failed drapes by error type; mesh generation time, triangle counts, simulation time, stopping frame, body-cloth and cloth-cloth collisions count – all per drape
- Rendering
    - Config: resolution, list of cameras
    - Stats: rendering time per drape

For all the information logged per drape, the summary statistics over the batch are provided.

Note: with the exception of triangle count, summary statistics per batch for the draping step are evaluated with the inclusion of the failed drapes as their purpose is to reflect the simulation process.

Data Statistics

Garment types

Summary of the garment types represented in the dataset. We identify five base categories: upper garments, pants, skirts, jumpsuits and dresses. Within each category, we identify the sub-categories for some common stylistic distinctions (asymmetry, sleeve length, bottoms length, presence of hood), as reported in the table below. Note that we chose the listed categories to be in line with related works and the garment types they report, however, it is not the only way to slice the dataset into garment types, and it does not account for many other stylistic choices present in the designs.

	Upper garments	**Skirts**	**Pants**	**Jumpsuits**	**Dresses**	**Total**
Total	34,140	29,648	10,177	7,575	44,333	
Asymmetric top	5,877	-	-	1,196	7,900	14,973
Sleeveless	10,341	-	-	2,327	13,499	26,167
Short sleeve	7,807	-	-	1,759	10,517	20,083
Long sleeve	15,992	-	-	3,489	20,317	39,798
Hoodie	3,067	-	-	671	3,918	7,656
Mini	-	8,986	1,267	1,112	5,876	27,011
Knee-length	-	7,092	3,004	2,578	8,909	23,272
Midi	-	4,627	2,273	1,812	7,856	16,693
Maxi	-	8,943	3,633	2,073	11,861	26,577

Panel counts

Here we report the basic distribution of the number of panel pieces in sewing patterns. Note that the number of panel pieces is the same for sewing patterns fitted to the neutral body shape and the corresponding patterns fitted to a random body shape.

Minimum	**Average**	**Maximum**
2	10.9	37

Triangle counts

Here is the basic distribution of the number of triangles across the garment meshes in the dataset. We define the mesh resolution by setting the average edge length to be 1 cm in the rest state, hence the number of triangles directly correlates with the amount of fabric used in a corresponding sewing pattern. The data is reported separately for drapes over neutral body shapes and drapes over random body shapes, since the triangle count depends on the sewing pattern sizes.

Body shapes	**Minimum**	**Average**	**Maximum**
Neutral body shape	330	26,363	187,972
Random body shape	364	26,462	177,100

Batch counts

Note that we share all the successful drapes in each subfolder, `./default_body/` and `./random_body/`, even if their corresponding pattern in the other subfolder failed. Hence, here we provide the counts of successful drapes for each batches' subfolders and a combined number – the number of drapes that were successful on both the neutral and the randomly generated body shapes.

Batch id	Neutral body	Random body	Both
Total	132,662	125,865	115,191
0	3,449	3,307	2,953
1	3,414	3,226	2,881
2	3,430	3,248	2,939
3	3,405	3,280	2,963
4	3,473	3,291	2,948
5	3,424	3,288	2,957
6	3,456	3,297	2,968
7	3,732	3,530	3,250
8	3,739	3,580	3,278
9	3,777	3,590	3,305
10	3,777	3,581	3,291
11	3,832	3,627	3,341
12	3,819	3,552	3,300
13	3,805	3,578	3,311
14	3,837	3,647	3,351
15	3,831	3,657	3,373
16	3,865	3,663	3,395
17	3,850	3,639	3,359
18	3,843	3,641	3,386
19	3,848	3,629	3,344
20	3,806	3,639	3,355
21	3,401	3,286	2,941
22	3,366	3,235	2,887
23	3,467	3,293	2,961
24	3,441	3,274	2,959
25	3,460	3,322	2,976
26	3,745	3,577	3,276
27	3,805	3,616	3,315
28	3,738	3,551	3,240
29	3,768	3,582	3,312
30	3,842	3,601	3,365
31	3,867	3,606	3,353
32	3,845	3,591	3,324
33	3,807	3,611	3,321
34	3,885	3,617	3,375
35	3,813	3,613	3,338