Project Note AP06-2019-01

OpenPose in the Public DGS Corpus

Authors: Marc Schulder, Thomas Hanke

Release: 9 December 2019

Contents

1 Introduction

One part of the second major release of the Public DGS Corpus (Konrad et al., 2019) is the addition of pose recognition data for all front-facing video recordings. It provides users with explicit machine-readable information on the location of various body parts, such as hands, shoulders, nose, ears, individual finger joints etc. Pose information was generated automatically using OpenPose (Simon et al., 2017; Cao et al., 2018). A visual representation of the computed information can be seen in Figure 1.

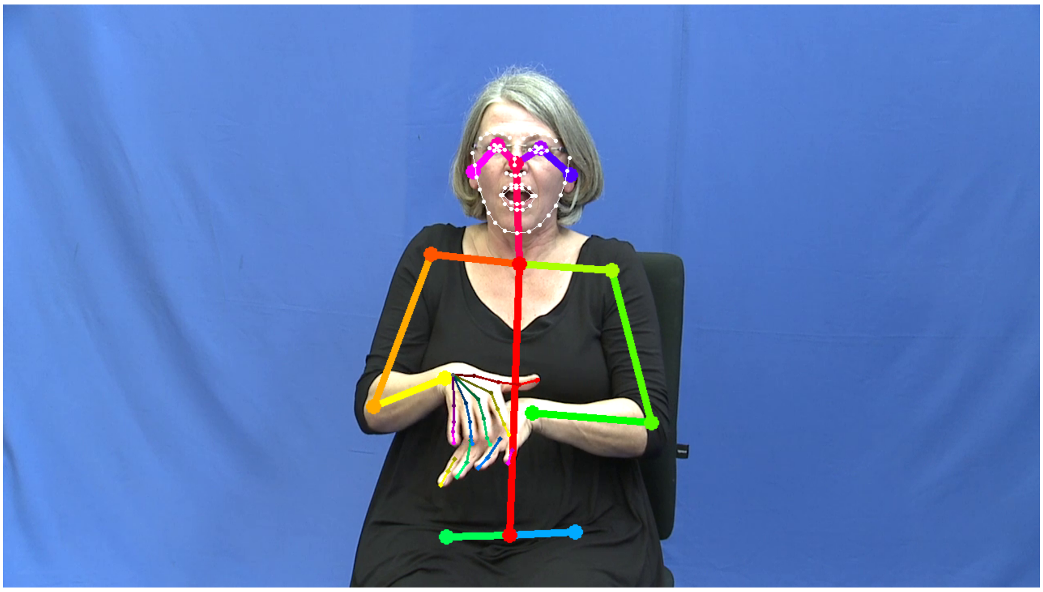
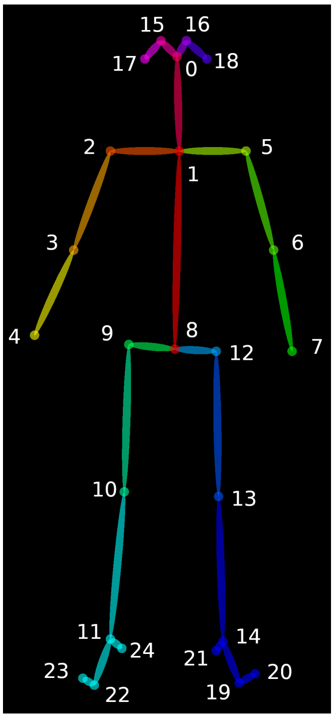


Figure 1: *Visual representation of the pose information provided by OpenPose, computed for a video from the DGS-Korpus project. Sets of keypoints are generated for the body, the face and each hand. Lines between the points are added to the visual representation to indicate the logical connection between individual keypoints.*

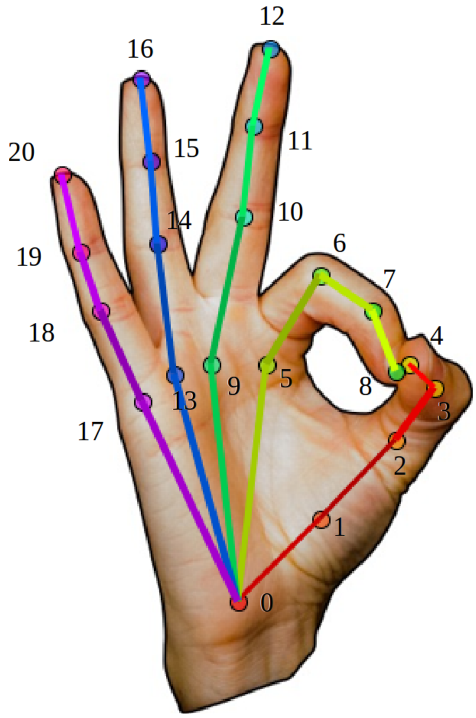
In Section 2 we provide an introduction to OpenPose, describing how it represents body poses in an image (Section 2.1) and how its data is formatted (Section 2.2). Section 3 describes how OpenPose is applied to the data of the Public DGS Corpus. We discuss processing the corpus (Section 3.1), applying anonymisation (Section 3.2), the format in which we provide the information (Section 3.3) and how it can be converted to the original OpenPose format (Section 3.4). Section 4 contains a brief reminder of the license conditions of the Public DGS Corpus.

2 OpenPose

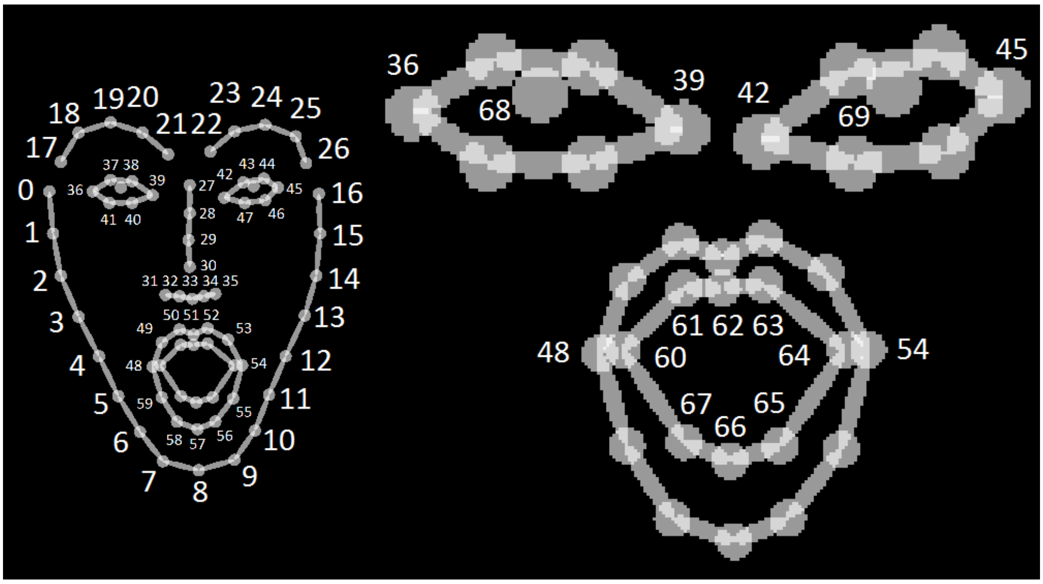
OpenPose is a pose estimation tool that determines the location of human bodies in an image, down to individual joints (shoulder, wrist, knee, etc.) and other points of interest like the nose and eyes. By applying OpenPose to frames of a video, movement estimation can be provided for individual body parts.



(a) *Body keypoints*



(b) *Hand keypoints*



(c) *Face keypoints*

Image source: https://github.com/CMU-Perceptual-Computing-Lab/openpose

Figure 2: *Skeleton maps of the keypoints determined by OpenPose. The number provided for each keypoint indicates its location in the output file (see Listing 1).*

In Section 2.1 we describe how the position of body parts is described through keypoints. Section 2.2 discusses the data format in which OpenPose outputs its predictions.

2.1 Keypoints

Each human body is represented by a number of **keypoints**. Each keypoint represents a skeletal joint or an otherwise relevant part of the body. For example, in the body model (shown in Figure 2a) the keypoint with index 0 represents the location of the nose, index 2 the right shoulder joint and index 8 the center of the hip.

The main OpenPose model by Cao et al. (2018) computes a general skeleton of the body, which identifies the joints in the shoulder, arms, hip and legs, as well as a few keypoints in the face and feet (see Figure 2a). Simon et al. (2017) add additional models to compute detailed hand and face models (see Figures 2b and 2c respectively).

In the case of the body model, two models are available: the default `BODY_25`, which provides 25 keypoints, and the `COCO` model, which reduces complexity of the hip and legs and provides 18 keypoints. We use the `BODY_25` model, which is recommended for most applications.

2.2 Data Format

For an example of the output format of OpenPose, see Listing 1. OpenPose creates an output file for each processed frame. Each file reports the OpenPose version used to generate the data and a list of people that were detected in the frame. Each person is represented as a mapping to eight lists, which represent 2D and 3D keypoints of the pose (i. e. the body), face, left hand and right hand. In our case only the 2D estimation was performed, so the four 3D keypoint lists are always empty.

A keypoint is represented as an x-coordinate, a y-coordinate and a confidence value. Accordingly, each keypoint list represents its n keypoints as a series of $3{*}n$ decimal numbers where the first three numbers represent the first keypoint, the fourth till sixth the second keypoint etc. The x- and y-coordinates specify pixel positions of the keypoint in the image.[1] Digits after the decimal point signify that the coordinate lies

[1] OpenPose also offers normalised keypoint values, but we choose to let the user decide whether such post-processing steps are desirable for their application.

Listing 1: *An abridged example of an OpenPose output file for a single frame containing one person. In this case, only 2-dimensional recognition was performed. Each line of numbers represents a single keypoint. A keypoint consists of an x-coordinate, a y-coordinate and a confidence value. Keypoints that were not visible in the frame are represented as "*`0,0,0`*". For reasons of space, this example omits many keypoints and shows "*`...`*" instead.*


```
{"version":1.2,
 "people":[
    {"pose_keypoints_2d":[
        662.642,184.701,0.844587,
        666.649,290.495,0.756731,
        545.137,304.218,0.569373,
        ...
        0,0,0,
        0,0,0],
     "face_keypoints_2d":[
        591.583,194.741,0.447692,
        ...
        670.916,155.074,0.909006],
     "hand_left_keypoints_2d":[
        760.857,472.859,0.385735,
        ...
        701.18,509.802,0.631602],
     "hand_right_keypoints_2d":[
        541.926,255.422,0.562895,
        ...
        596.183,166.286,0.624597],
     "pose_keypoints_3d":[],
     "face_keypoints_3d":[],
     "hand_left_keypoints_3d":[],
     "hand_right_keypoints_3d":[]
    }
  ]
}
```

Transcript	Age Group	Format	Topics	iLex File	ELAN File	Movie A	Movie B	Movie Total	SRT File	Movie AB	OpenPose
dgskorpus_ber_01	18-30	Experience of Deaf Individuals	Communication: Communication with Hearing People Deaf Culture						CC		

Figure 3: *File selection on ling.meine-dgs.de. OpenPose data for each transcript is offered in the final column.*

between two pixels. The x-coordinate goes from left to right and the y-coordinate from top to bottom. The confidence value represents the classifier's confidence that the keypoint is positioned correctly. It is given as a decimal value between 0 and 1. If a keypoint could not be detected (e. g. because the body part in question is not visible in the image) its three components are all set to 0.

For an example, observe the frame output for a video with 1280×720 pixels video in Listing 1. It was computed with OpenPose version 1.2 and detected a single person. The body gives the location of the nose (keypoint index 0, see Figure 2a) as being at pixel position 662.642×184.701. This means the nose is almost at the horizontal center and 25% below the top of the image. OpenPose has a high confidence of 84.4587% that this position is correct. The final keypoints of the body model (right toe and right ankle, indices 23 and 24), on the other hand, are marked as missing as they are not visible in the picture.

3 OpenPose on the Public DGS Corpus

We computed poses for all transcripts of the Public DGS Corpus. For each transcript, the front-facing recordings of the informants (*Movie A* and *Movie B*) were processed. We compute keypoints for the body, face and both hands.

The resulting pose information is available on ling.meine-dgs.de as part of the files offered for each transcript (see Figure 3). For each transcript there exists a single file containing all its OpenPose data. The file is contained in a compiled `gzip` archive to reduce its download size.

The remainder of this section provides information specific to the Public DGS Corpus release of OpenPose data. Section 3.1 discusses how OpenPose was run to process the corpus data. Section 3.2 details how sections that are anonymised in the public release were handled.

Figure 4: *Anonymised utterance in the Public DGS Corpus. Both the right hand and mouth are anonymised to obfuscate both sign and mouthing. In other cases, only one or the other or the entire body may be obfuscated.*

Section 3.3 introduces the format of the data wrapper which bundles all frames of a transcript. In Section 3.4 we discuss how the original OpenPose frame data can be extracted from the data wrapper.

3.1 Processing the corpus

OpenPose was run on the original full-size recordings of the DGS-Korpus project. Technical details on how OpenPose was run on a high performance cluster to process the corpus data, see Hanke (2018).

The resulting data was reviewed for dropped frames and erroneous detections of too many bodies. As the front-facing cameras record a single person, detections of multiple bodies were almost always in error, except for a few cases where another participant leaned into the image. These false positives were subsequently removed from the data.

3.2 Anonymisation

The Public DGS Corpus has been anonymised to protect personal information, such as names of people (Bleicken et al., 2016). In the video data this is achieved by drawing black rectangles over the parts of the video that require anonymisation. An example of this can be seen in Figure 4.

Pose recognition for the corpus was performed on the original recordings, so anonymisation is applied as a post-processing step. We differentiate between two cases:

1. Mouth anonymisation.
2. Complete anonymisation.

In the case of mouth anonymisation we remove all face model keypoints of the given frame. For complete anonymisation, we remove the keypoints of all four models, effectively removing all pose information for the frame.[2]

If a model is anonymised, its list of keypoints is replaced with an empty list, just like the 3D models in Listing 1. This allows the user to differentiate between anonymised data and naturally invisible keypoints, which are shown as an explicit triple of zeroes, rather than as missing data.

3.3 Data Format

By default, OpenPose generates one file per frame (see Section 2.2). As we cover many hours of recordings, this would result in a huge number of very small files. This is not advisable, as file systems can be markedly slower at handling them compared to small numbers of large files.

Instead, we create a wrapper format that collects all frames for all recordings of a transcript. That means one file contains pose information for both *Movie A* and *Movie B* (see Figure 3). For each recording we provide relevant metadata in addition to the OpenPose frame output. A description of the top-level fields of a recording is given in Table 1. An example of the resulting wrapper file can be seen in Listing 2.

[2] Complete anonymisation also covers cases in which a single hand was anonymised. This is because the area obfuscated is usually larger, covering the arm and possibly shoulder and face. In such cases we choose to err on the side of caution and remove all pose information, rather than reveal keypoints that annotators intended to anonymise.

id	The ID of the transcript that the recording belongs to.
camera	The camera perspective that the recording represents. `a1` for the front-facing recording of informant A and `b1` for the front-facing recording of informant B.
width	The pixel width of the recording.
height	The pixel height of the recording.
frames	A key-value mapping from frame index (starting at 0) to the complete OpenPose output of that frame (see Section 2.2). The mapping lists all frames of the associated video file.

Table 1: *The top-level fields of each recording entry of the wrapper format. The fields* id, camera, width *and* height *are recording metadata applicable to each frame. For an example of how these fields are filled, see Listing 2.*

Listing 2: *An abridged example of pose information for the corpus transcript with the ID* 1413451-11105600-11163240*, containing two recordings. Frame information is omitted here (marked as "`...`"), but an example can be seen in Listing 1.*

```
[ {
    "id": "1413451-11105600-11163240",
    "camera": "a1",
    "width": 1280,
    "height": 720,
    "frames": {
      "0": { ... },
      "1": { ... },
      ...
      "16840": { ... }
    }
  },
  {
    "id": "1413451-11105600-11163240",
    "camera": "b1",
    "width": 1280,
    "height": 720,
    "frames": { ... }
} ]
```

3.4 Conversion to Single Frame Format

To convert the data from the wrapper format described in Section 3.3 back to the standard OpenPose output (Section 2.2), we provide a Python converter script.[3] It requires at least Python 2.7 or Python 3 to run.

The script takes one or several wrapper files as input and for each input file creates a directory of frame files. The name of the output directory matches the name of the input file, unless an alternative directory is provided. To retain the additional metadata of the wrapper, the filenames of the created frame files follow the following naming pattern:

`**id**_**camera**.**width**x**height**.frame_**frame**.keypoints.json`

Words in bold represent variables that are filled with the five metadata values from Section 3.3. In the case of **`frame`**, the frame index is given as a 13-digit value with prepended zeroes, matching the output of OpenPose.

For example, applying the conversion script to the pose file for transcript *1413451-11105600-11163240* shown in Listing 2 would result in a directory containing 33,680 files, half of which represent the frames from *Movie A* and half those of *Movie B*. The very first file in the directory would be the first frame of *Movie A* and have the following filename (split into two lines for reasons of space):

```
1413451-11105600-11163240_a1.1289x720.
frame_0000000000000.keypoints.json
```

The filename for the second frame would be identical to the first, except it would say `frame_0000000000001` instead of `frame_0000000000000`.

4 Licence

The OpenPose data of the Public DGS Corpus falls under the same licence and usage restrictions as the remaining parts of the Public DGS Corpus.[4] It may only be downloaded and used for linguistic research. Should you publish research based on the material, please cite the corresponding publications.[5]

[3] https://github.com/DGS-Korpus/Public-Corpus-OpenPose-frame-extractor
[4] http://ling.meine-dgs.de/license_en.html
[5] http://dgs-korpus.de/index.php/publications.html

References

Bleicken, Julian et al. (2016). “Using a Language Technology Infrastructure for German in order to Anonymize German Sign Language Corpus Data”. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. Portorož, Slovenia, pp. 3303–3306. URL: https://www.aclweb.org/anthology/L16-1526.pdf.

Cao, Zhe et al. (2018). *OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields*. Preprint. arXiv: 1812.08008 [cs.CV].

Hanke, Thomas (2018). *AP04-2018-01. Processing DGS-Korpus Data with OpenPose on the Hamburg High Performance Cluster*. Project Note. Universität Hamburg. URL: http://www.dgs-korpus.de/arbeitspapiere/AP04-2018-01.html.

Konrad, Reiner et al. (2019). *MY DGS – Annotated. Public Corpus of German Sign Language, 2nd Release*. Dataset. DGS-Korpus project, IDGS, Hamburg University. DOI: 10.25592/dgs.corpus-2.0.

Simon, Tomas et al. (2017). “Hand Keypoint Detection in Single Images Using Multiview Bootstrapping”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, Hawaii, USA, pp. 4645–4653. DOI: 10/ggd43g.