

---

## Motion Representations for Articulated Animation

This repository contains the source code for the CVPR'2021 paper Motion Representations for Articulated Animation by Aliaksandr Siarohin, Oliver Woodford, Jian Ren, Menglei Chai and Sergey Tulyakov.

For more qualitative examples visit our project page.

### Example animation

Here is an example of several images produced by our method. In the first column the driving video is shown. For the remaining columns the top image is animated by using motions extracted from the driving.



### Installation

We support `python3`. To install the dependencies run:

```
1 pip install -r requirements.txt
```

### YAML configs

There are several configuration files one for each `dataset` in the `config` folder named as `config/dataset_name.yaml`. See `config/dataset.yaml` to get the description of each parameter.

See description of the parameters in the `config/vox256.yaml`. We adjust the the configuration to run on 1 V100 GPU, training on 256x256 dataset takes approximately 2 days.

---

## Pre-trained checkpoints

Checkpoints can be found in [https://drive.google.com/drive/folders/1jCeFPqfU\\_wKNYwof0ONICwsj3xHlr\\_tb?usp=sha](https://drive.google.com/drive/folders/1jCeFPqfU_wKNYwof0ONICwsj3xHlr_tb?usp=sha)

## Animation Demo

To run a demo, download a checkpoint and run the following command:

```
1 python demo.py --config config/dataset_name.yaml --driving_video path/to/driving --source_image path/to/source --checkpoint path/to/checkpoint
```

The result will be stored in `result.mp4`. To use Animation via Disentaglemet add `--mode avd`, for standard animation add `--mode standard` instead.

## Colab Demo

We prepared a demo runnable in google-colab, see: [demo.ipynb](#).

## Training

To train a model run:

```
1 CUDA_VISIBLE_DEVICES=0 python run.py --config config/dataset_name.yaml --device_ids 0
```

The code will create a folder in the log directory (each run will create a time-stamped new folder). Checkpoints will be saved to this folder. To check the loss values during training see `log.txt`. You can also check training data reconstructions in the `train-vis` subfolder. Then to train **Animation via disentanglement (AVD)** use:

```
1 CUDA_VISIBLE_DEVICES=0 python run.py --checkpoint log/{folder}/cpk.pth --config config/dataset_name.yaml --device_ids 0 --mode train_avd
```

Where `{folder}` is the name of the folder created in the previous step. (Note: use backslash " before space.) This will use the same folder where checkpoint was previously stored. It will create a new checkpoint containing all the previous models and the trained `avd_network`. You can monitor performance in log file and visualizations in `train-vis` folder.

---

## Evaluation on video reconstruction

To evaluate the reconstruction performance run:

```
1 CUDA_VISIBLE_DEVICES=0 python run.py --config config/dataset_name.yaml
  --mode reconstruction --checkpoint log/{folder}/cpk.pth
```

Where {`folder`} is the name of the folder created in the previous step. (Note: use backslash “ before space.) The `reconstruction` subfolder will be created in the checkpoint folder. The generated video will be stored to this folder, also generated videos will be stored in `png` subfolder in loss-less ‘png’ format for evaluation. Instructions for computing metrics from the paper can be found [here](#).

## TED dataset

For obtaining TED dataset run the following commands:

```
1 git clone https://github.com/AliaksandrSiarohin/video-preprocessing
2 cd video-preprocessing
3 python load_videos.py --metadata ../data/ted384-metadata.csv --format .
  mp4 --out_folder ../data/TED384-v2 --workers 8 --image_shape 384,384
```

## Training on your own dataset

- 1) Resize all the videos to the same size, e.g 256x256, the videos can be in ‘.gif’, ‘.mp4’ or folder with images. We recommend the latter, for each video make a separate folder with all the frames in ‘.png’ format. This format is loss-less, and it has better i/o performance.
- 2) Create a folder `data/dataset_name` with 2 subfolders `train` and `test`, put training videos in the `train` and testing in the `test`.
- 3) Create a config file `config/dataset_name.yaml`. See description of the parameters in the `config/vox256.yaml`. Specify the dataset root in `dataset_params` specify by setting `root_dir: data/dataset_name`. Adjust other parameters as desired, such as the number of epochs for example. Specify `id_sampling: False` if you do not want to use `id_sampling`.

## Additional notes Citation:

```
1 @inproceedings{siarohin2021motion,
2   author={Siarohin, Aliaksandr and Woodford, Oliver and Ren, Jian
  and Chai, Menglei and Tulyakov, Sergey},
```

---

```
3     title={Motion Representations for Articulated Animation},
4     booktitle = {CVPR},
5     year = {2021}
6 }
```