Processing DGS-Korpus Data with OpenPose on the Hamburg High Performance Cluster

Edited by: Thomas Hanke, 2018-12-31
Thomas Hanke, 2019-02-02, replaced screendumps with those showing actual contents

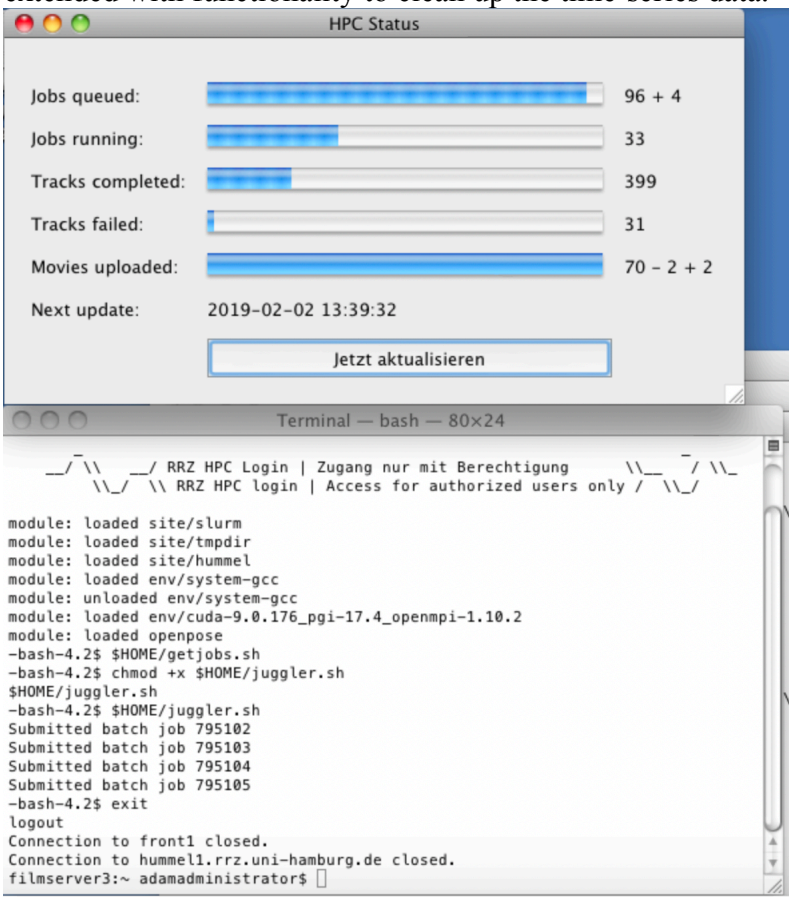
Introduction

OpenPose is very promising to enable accurate motion analysis on the DGS-Korpus data. For each frame of a video, OpenPose computes the number of persons visible, and for each person it determines pixel coordinates of a set of body joints. The 2018 edition of OpenPose adds coordinates on the face as well on the hands.
As OpenPose is very computationally heavy, there is no chance to compute even a representative sample from the corpus on a single desktop computer. Instead, we have decided to use the University of Hamburg's High Performance Cluster (HPC). The HPC's GPU partition consists of 54 nodes with two GPUs each. Within the job limit of twelve hours, a single node can compute 100'000 frames from a DGS-Korpus A1 or B1 video, or 50'000 frames from a C video. As the video footage of one recording session is around 1M frames per camera, it takes a single node approx. 20 days to compute the videos of one recording session. So the total computation time for the corpus data is in the order of magnitude of 165*20 days = 9 years. If we assume that on average 30 GPU nodes are available to the project, the computation will take approx. 110 days or almost four months.
It is therefore essential to keep the HPC busy, i.e. to have at any time as many jobs queued as nodes can become available. In order to accomplish this, jobs query their target video and frame range from a database when launched, and signal success or failure via an update to the same database. Precautions have been taken that the database server can handle 165*(1M+1M+3M) frame data records consisting of json data.

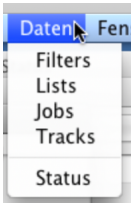
OpenPoseJuggler

OpenPoseJuggler was developed to monitor the HPC OpenPose project, to submit new jobs to the cluster and to upload videos to be processed to the cluster work space, and to purge them when they have been completely processed (job queue max length 100, work space size 500GB).
Every hour, OpenPoseJuggler, itself running on one of the project's server machines, connects to the HPC via ssh to read the job queue and queries the database to determine progress. It then submits new jobs if others have terminated, purges videos and starts uploading new ones. In theory, this should allow the OpenPose project to run without human intervention, even in the case that the HPC is temporarily shut down.
In addition to the monitor function, OpenPoseJuggler also provides access to the database tables used for the project with customizable views as known from iLex. This gives the user the possibility to analyse jobs that failed, and to re-schedule tracks once the failure reason has

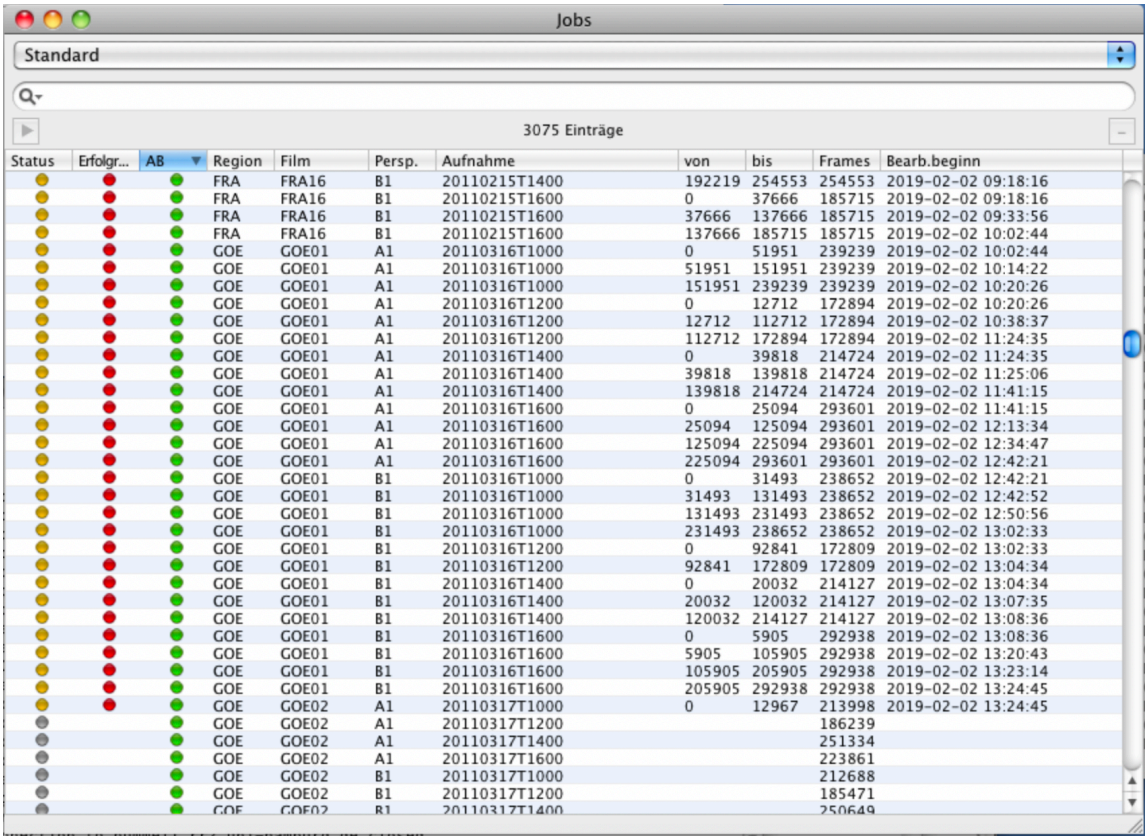
been eliminated. Once a good part of data has been processed, OpenPoseJuggler will be extended with functionality to clean up the time-series data.



Screen 1: Monitor window and Terminal window opened by the monitor

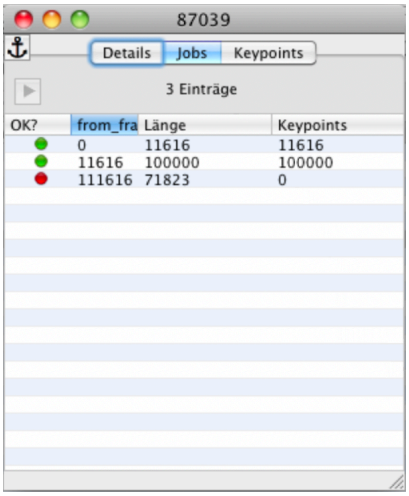


Screen 2: Database tables accessible through OpenPoseJuggler



Status	Erfolgr...	AB	Region	Film	Persp.	Aufnahme	von	bis	Frames	Bearb.beginn
			FRA	FRA16	B1	20110215T1400	192219	254553	254553	2019-02-02 09:18:16
			FRA	FRA16	B1	20110215T1600	0	37666	185715	2019-02-02 09:18:16
			FRA	FRA16	B1	20110215T1600	37666	137666	185715	2019-02-02 09:33:56
			FRA	FRA16	B1	20110215T1600	137666	185715	185715	2019-02-02 10:02:44
			GOE	GOE01	A1	20110316T1000	0	51951	239239	2019-02-02 10:02:44
			GOE	GOE01	A1	20110316T1000	51951	151951	239239	2019-02-02 10:14:22
			GOE	GOE01	A1	20110316T1000	151951	239239	239239	2019-02-02 10:20:26
			GOE	GOE01	A1	20110316T1200	0	12712	172894	2019-02-02 10:20:26
			GOE	GOE01	A1	20110316T1200	12712	112712	172894	2019-02-02 10:38:37
			GOE	GOE01	A1	20110316T1200	112712	172894	172894	2019-02-02 11:24:35
			GOE	GOE01	A1	20110316T1400	0	39818	214724	2019-02-02 11:24:35
			GOE	GOE01	A1	20110316T1400	39818	139818	214724	2019-02-02 11:25:06
			GOE	GOE01	A1	20110316T1400	139818	214724	214724	2019-02-02 11:41:15
			GOE	GOE01	A1	20110316T1600	0	25094	293601	2019-02-02 11:41:15
			GOE	GOE01	A1	20110316T1600	25094	125094	293601	2019-02-02 12:13:34
			GOE	GOE01	A1	20110316T1600	125094	225094	293601	2019-02-02 12:34:47
			GOE	GOE01	A1	20110316T1600	225094	293601	293601	2019-02-02 12:42:21
			GOE	GOE01	B1	20110316T1000	0	31493	238652	2019-02-02 12:42:21
			GOE	GOE01	B1	20110316T1000	31493	131493	238652	2019-02-02 12:42:52
			GOE	GOE01	B1	20110316T1000	131493	231493	238652	2019-02-02 12:50:56
			GOE	GOE01	B1	20110316T1000	231493	238652	238652	2019-02-02 13:02:33
			GOE	GOE01	B1	20110316T1200	0	92841	172809	2019-02-02 13:02:33
			GOE	GOE01	B1	20110316T1200	92841	172809	172809	2019-02-02 13:04:34
			GOE	GOE01	B1	20110316T1400	0	20032	214127	2019-02-02 13:04:34
			GOE	GOE01	B1	20110316T1400	20032	120032	214127	2019-02-02 13:07:35
			GOE	GOE01	B1	20110316T1400	120032	214127	214127	2019-02-02 13:08:36
			GOE	GOE01	B1	20110316T1600	0	5905	292938	2019-02-02 13:08:36
			GOE	GOE01	B1	20110316T1600	5905	105905	292938	2019-02-02 13:20:43
			GOE	GOE01	B1	20110316T1600	105905	205905	292938	2019-02-02 13:23:14
			GOE	GOE01	B1	20110316T1600	205905	292938	292938	2019-02-02 13:24:45
			GOE	GOE02	A1	20110317T1000	0	12967	213998	2019-02-02 13:24:45
			GOE	GOE02	A1	20110317T1200			186239	
			GOE	GOE02	A1	20110317T1400			251334	
			GOE	GOE02	A1	20110317T1600			223861	
			GOE	GOE02	B1	20110317T1000			212688	
			GOE	GOE02	B1	20110317T1200			185471	
			GOE	GOE02	B1	20110317T1400			250649	

Screen 3: Jobs table



OK?	from_fra	Länge	Keypoints
	0	11616	11616
	11616	100000	100000
	111616	71823	0

Screen 4: Tracks detail window with the corresponding jobs listed

Availability

OpenPoseJuggler source code as well as the Python and bash scripts are available to other researchers upon request. While it has been tried under macOS only, it should also run under Windows and Unix. The scripts have been written to talk to a PostgreSQL database. As the database interactions are rather straight-forward, a switch to another database machine should be possible with minimal effort, provided that the database allows global serializability. Other dependencies on specific functionality of the Hamburg HPC cluster are encapsulated in a few places. The only script actually accessing the iLex database is the SQL query to determine which videos are to be processed, providing their pathes and durations. (Alternatively, durations can also easily be computed using ffprobe.) Obviously, OpenPoseJuggler needs to

run on a machine that has access to the video pathes in order to be able to upload them. In order not to have to store the ssh password in a local copy of OpenPoseJuggler, we recommend to configure ssh access to use certificates.