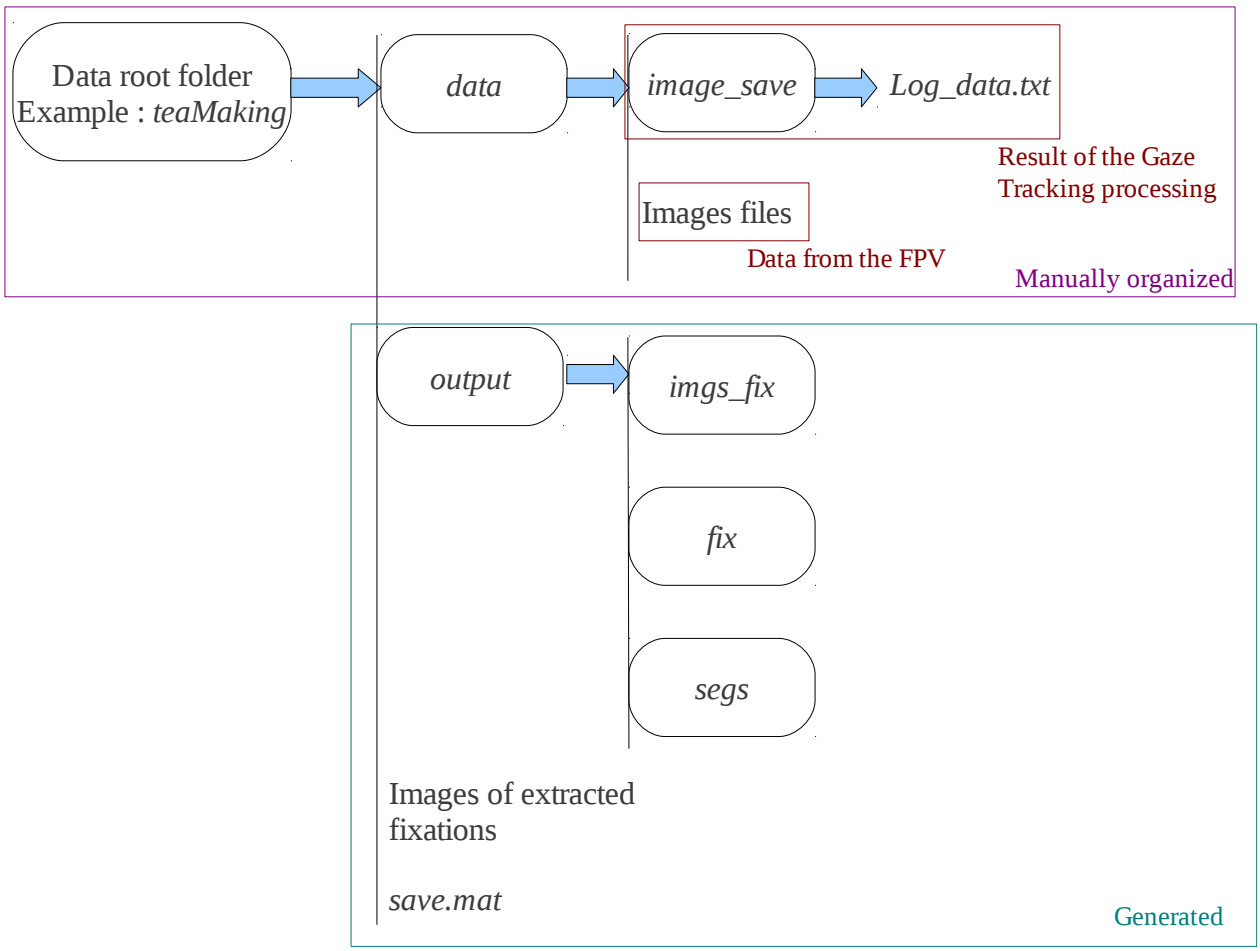
**Preliminaries :**

- Compile the two executables in *tracking*. (Running *cmake* .. then *make* in *tracking/bin* should be enough, it is only using OpenCV)
- Compile the segmentation executable in *fixationSegment/bin* (see later for the CUDA support)

**Folder structure for each video :**




**How to process a video?**

- Create the “Manually organized” part.
- Run *./removeSaccades PATH_TO_ROOT_FOLDER*
  - The algorithm call *trackGaze* to remove the environment movement
  - Detect fixations and show the results
  - Call *trackFixations* to track the found points on severals images
  - Subsample the videos where points are found, copy the found images in the root directory, copy the images with the fixations points drawn in *imgs_fix* and store all the informations in *save.mat*
- Run *./fixation2.sh PATH_TO_ROOT_FOLDER N*
  - Where N=0,1 or 2. 0 is for only using the fixation points. 1 to use a small pattern around it and 2 a bigger pattern.

    - This is gonna create the .fix files in the *fix* directory. These files are used for segmentation.
- Run *./run_segment.sh PATH_TO_ROOT_FOLDER N*
    - Where N is the number of threads (Matlab instances)
    - This is calling the *segment* executable for segmenting every image in the root folder according to the corresponding .fix file.
    - The resulting segments are in *segs.*

Tip : you can process the removeSaccades part on your own computer (this is not so long) then copy only the images extracted and the save.mat to a more powerful machine to do the segmentation part.

**Modifying the input data :**
If you want to modify the input data, you have to :

- put the images of your videos in the *data* folder
- modify the macro `IMAGE_BASENAME` in the beginning of *fixationSegment/trackingGaze.cpp and fixationSegment/trackingFixations.cpp*
- modify the `imageBasename` variable in the beginning of *segmentation/removeSaccades.m*
- modify the few lines following (reading of the *Log_data.txt)* to read instead your new data. What's need to be set is only the *fixs* variable.

**Compiling the CUDA version of the segmentation algorithm**
**TODO**

**Doing edgeDetection on buyukada and segmentation on shona**
You have to run the script on shona with the executable already compiled on buyukada.
In the middle of *seg.m* you can choose between two different calls of *segment.* One of them will first call *segmentShonaBuyukada.sh* which is a script which calls via ssh the computation of the edgeboundary on buyukada. Buyukada will directly write on shona in a folder which would have been remotely mounted on buyukada (via sshfs for instance)
TO BE IMPROVED

**NOTES**
In the dispersionExtraction function in removeSaccades, there are two paramaters you can modify which are the dispersionThreshold and the durationThreshold. You may want to reduce the dispersionThreshold if the data is not shaky to improve results.
In the fixationSegment/segment.cpp, images are resized prior to segmentation to improve speed, you can choose the sizes (for GPU or CPU) at the beginning of the file.
Actually the tracking (for gaze and fixations) is very simple (Lukas Kanade). You may want to improve that in the fixations tracking so that the points are not fleeing away from the objetcs sometimes.