**SPM@Unipi-2016/2017**

**Project Report [Video Filtering]**

By: Ahmad Alleboudy

**What has been implemented in the project**

1. A pipeline using fastflow to apply a filter[using OpenCV] on an input video and produce the output result.[parallel.cpp]
2. Same filtering implemented in a sequential manner [sequential.cpp]
3. Same filtering implemented using C++11 promises [threaded.cpp]

**Possible major design choices**

1. A fast flow farm with a custom emitter and collector [would need to worry about the frames ordering]
2. A fast flow Ordered Farm with a custom emitter and collector [would not have to worry about the frames ordering as it is already handled by the ordered farm]

I settled with the ordered farm, also implemented a small buffering mechanism at the collector to defer the output flushing. [small issue, OpenCV doesn’t allow flushing multiple frames at the same time, and per my knowledge about fast flow the parallel for requires independent iterations thus the order is not guaranteed so, this is just deferring the output flushing ]

**How to compile**

Will need to have OpenCV installed and the fastflow root folder present on the machine

In the project repository main folder run the build.sh file or type in a terminal the following:

g++ -I$FF\_ROOT -I/usr/local/include -I/usr/local/include/opencv/ -I/usr/local/include/opencv2 -L/usr/lib/x86\_64-linux-gnu/ -g -o run parallel.cpp -lopencv\_videoio -lopencv\_core -std=c++11 -pthread

Where $FF\_ROOT is where the fastflow root folder

“run” is the name of the resulting compiled binary

“parallel.cpp” is the parallel implementation [could also do sequential.cpp or threaded.cpp instead]

other imports are for the openCV

**To run the compiled Binary type**

For the parallel.cpp

./run path/to/input/video/file.mp4 path/to/output/video/file.mp4 NumberOfWorkersForTheFarm SizeOfOutputBuffer

path/to/input/video/file.mp4 is where the input video resides on the desk

path/to/output/video/file.mp4 is where we wish to have the output video

NumberOfWorkersForTheFarm is the degree of parallelism of the farm

SizeOfOutputBuffer number of frames allowed in the output buffer before flushing the buffer

For the sequential.cpp

/run path/to/input/video/file.mp4 path/to/output/video/file.mp4

For the threaded.cpp

./run path/to/input/video/file.mp4 path/to/output/video/file.mp4 NumberOfWorkers

path/to/input/video/file.mp4 is where the input video resides on the desk

path/to/output/video/file.mp4 is where we wish to have the output video

NumberOfWorkers is the number of threads to use