## **Master ST Computer Science option ISM**

UE VNAC

University Bordeaux Faculty S&T academic year 2016 – 2017

## Part 1. Lab 1 (4 hours, evaluated)

## **Introduction to OpenCV**

Purpose: Code and compare various motion estimation algorithms using C++ API of OpenCV.

**Work environment**: Linux, C++, OpenCV, ffmpeg, Gnuplot.

## 1. Introduction to OpenCV

OpenCV documentation is available here: http://docs.opencv.org/

Get video files from: http://dept-info.labri.fr/~mansenca/VNAC2016/MotionEstimationDataSet/

Get a basic CMakeLists.txt from here: http://dept-info.labri.fr/~mansenca/VNAC2016/

*Using* C++ API of OpenCV, code a program that does the following:

1.1) Loads a movie file given as parameter, displays its frames, save its frames to disk. - first work on RGB frames than work only with Y plane of VCrCh frames

- then work only with Y plane of YCrCb frames

*1.2) Computes the following measures: MSE, PSNR, Entropy, and the Error Image* - between two consecutive frames: I(t) et I(t-1)

- between two frames with temporal distance  $\Delta t$  (given as parameter): I(t) et I(t- $\Delta t$ )

We consider images are of size M\*N. Here are the various formulae:

Mean Square Error (MSE):  $MSE = \frac{1}{N * M} \cdot \sum (I(p,t) - I(p,t-\Delta t))^2$ 

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right)$$

Peak Signal-to-Noise Ratio (PSNR):

Entropy: 
$$Ent(X) = -\sum p(x_i) \log_2 p(x_i)$$

The error image:  $E(p,t) = min(255, max(0, I(p,t) - I(p,t-\Delta t) + 128))$ 

1.3) Add an option to the program to save these measures and plot them with **gnuplot**.

Send an archive with your code and pictures of measures curves, named MYNAME\_Lab1.tar.bz2, to : boris.mansencal@labri.fr