

Lab 1
(2h00, evaluated)

Optical Flow estimation & Global motion estimation in the image plane with RANSAC algorithm

Purpose: Study and measure different metrics of information content of motion image sequence: MSE, PSNR and Entropy. Practice Optical Flow estimation methods with OpenCV
Compute Global motion estimation and visualize the distribution of the energy of residual motion.

Work environment: Linux, Python, Numpy, OpenCV.

Get skeleton code here:

/net/ens/DeepLearning/DLCV2022/lab1_skeleton.py
or http://dept-info.labri.fr/~mansenca/DLCV2022/lab1_skeleton.py

Video files are here:

/net/ens/DeepLearning/DLCV2022/lab1_data/

1. Metrics computation

The provided *lab1* python code allows to load a video file, to compute the optical flow between two frames with temporal distance Δt : $I(t)$ and $I(t-\Delta t)$, and visualize the optical flow.

1.1 Run the code

First you need to do:

source </net/ens/DeepLearning/python3/tensorflow2/bin/activate>

Run the code on various video files (in particular **vtest.avi** and **PersonConvergence_720.MOV**).

1.2 Code the various metrics

We want to study and measure different metrics of information content of motion image sequence:

- **MSE** (Mean Square Error) between frames: characterizes contrasts and motion between sequences. Can be strong in case of noisy video content.
- **PSNR** (Peak Signal to Noise Ratio): computed on the basis of MSE, characterizes the difference between frames of a motion image sequence. It is used for assessment of motion compensation and coding methods.
- **Entropy**: the measure of amount of information in a system. We study the entropy of an original motion picture sequence and of the error sequence which is the difference between consecutive frames of a motion image sequence.

We consider images are of size $M \times N$.

Here are the various formulae:

Mean Square Error (MSE) :
$$MSE = \frac{1}{N \times M} \cdot \sum (I(p, t) - I(p, t - \Delta t))^2$$
 (where p is a point/pixel coordinate in the image)

Peak Signal-to-Noise Ratio (PSNR):
$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right)$$

Entropy:
$$Ent(X) = - \sum p(x_i) \log_2 p(x_i)$$
 (where $p(x_i)$ is the probability to have the color x_i)

Code these metrics and save their plots for at least two videos.

We also want to compute the error image: $E(p, t) = I(p, t) - I(p, t - \Delta t)$

and the displayable error image: $E(p, t) = \min(255, \max(0, I(p, t) - I(p, t - \Delta t) + 128))$ (where p is a point/pixel coordinate in the image)

2. Global Motion Estimation

You have to estimate the global motion, centered at the center of the image, from the motion vectors obtained with the provided motion estimation method.

With the help of OpenCV, you will first compute the perspective transformation, or homography, with the least square method coupled with the RANSAC algorithm, and then compute the global motion estimation.

You can then compute the residual motion and visualize the energy of the residual motion, that is the normalized L2 distance between the original motion vectors and the computed global motion vectors.

You must send an archive with your code and report.

The report should contain the curves for the three metrics and some images (error image, energy of residual motion) for a chosen frame for two videos, including

PersonConvergence_720.MOV. You must discuss the curves.

This archive must be named [LASTNAME_firstname_Lab1_DLCV.tar.bz2](#) and sent to: boris.mansencal@labri.fr with DLCV, your group, name and lab number in the e-mail subject: for example “[DLCV-mansencal: DOE_John_Lab1](#)”.