Optical Flow Estimation and Global Motion Estimation in the image plane with RANSAC algorithm

Awet Haileslassie Gebrehiwot, Santiago Herrero Melo

I. INTRODUCTION

In this first lab session of *Deep Learning in Computer Vision* we will study and measure different metrics of information content of motion image sequence, practice Optical Flow estimation methods with OpenCV, compute Global Motion Estimation and visualize the distribution of the energy of residual motion in the different video sequences. The project will be developed in python language, using the tensorflow environment. The libraries *numpy* (for mathematical operations in matrices), *cv2* (for using OpenCV) and *matplotlib* (for 2D graphs plotting) will be used.

II. DATA

Although it is only required to perform the implementation of the code with two video sequences, the dataset is composed of eight different video sequences.

- *Birds_720* shows a river in which the main movement we see is a small bird starting to fly. Also, there is camera motion following this bird.
- *BoatSLow_720* permanently shows a boat moving slowly, but also cars that move fast in both directions. There is a small camera motion following the boat.
- *PersonConvergence_720* shows a person walking straight along a river, going away from the camera, which is fixed.
- *SmallCars_720* shows some cars moving along a road recorded from far and with occlusion. The camera is not properly fixed so there is small movement.
- *v_LongJump_g13_c06* shows a person running and finally performing a jump. While the person is running, the camera follows it.
- *v_SkateBoarding_g25_c01* shows a man skateboarding, and the camera is following him from a fixed distance.
- *v_TaiChi_g16_c04* shows a person doing some taichi movements, recorded with a fixed camera.
- *vtest* shows a street with different people walking on it, recorded from a fixed camera.

III. IMPLEMENTATION

Motion estimation is the process of determining motion vectors that describe the transformation from one frame to another, usually it is done from adjacent frames in a video sequence. In the case of the proposed code, there is a variable *deltaT* which determines the distance between the two frames from which we calculate this transformation.

A. MSE

MSE (Mean Square Error) between frames: it is a measure of quality of estimation that characterizes contrasts and motion between sequences.

$$MSE = [\sum [I[p,t] - I[p,t-\Delta t]]^2] * \frac{1}{N*M} \quad (1)$$

Where N and M are the dimensions of the frame I, I[p,t] is the current frame, and I[p,t-Δ t] is the revious frame.

B. PSNR

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. It shows the ratio between the possible power of a signal and the power of its corrupting noise.

$$PNSR = 10 * \log \frac{255^2}{MSE} \quad (2)$$

C. ENTROPY

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.

$$Ent(X) = -\sum p(x_i) \log_2 p(x_i) \quad (3)$$

Where p(x_i) is the probability of finding one specific value between all the pixels of the image. This value is very easily obtained calculating the histogram of the image, with the function *cv2.calcHist*.
We also obtain the error image as the difference between the two consecutive frames.

$$ErrorImage: \;\; E(p,t) = I(p,t) - I(p,t-\Delta t) \quad (4)$$

D. Optical Flow

The optical flow shows the direction of the movement between the two consecutive frames. This value was already provided, using the function *cv2.calcOpticalFlowFarneback*.

E. Compensated Frame

It is a remapping of the frame, from which we should obtain less errors. The calculation of this compensated frame is also provided in the original code.

F. GME

Global motions Estimation in a video sequence is the process of estimating the transform parameter caused by camera motion.
In order to calculate this value, we first have to compute the homography matrix from the source and destiny points of the consecutive frames. For doing so, we will use the function *cv2.findHomography*. For this calculation, we will use the RANSAC algorithm, which is an iterative method used in data that contains outliers when outliers are to be accorded no influence on the values of the estimates, which means it can be interpreted as an outlier detection method.
This homography will then be used to compute the final GME with the function *cv2.perspectiveTransform*.
Finally, we can compute the error related to the GME by applying the normalized L2 distance between the original motion vectors and the computed global motion vectors.

G. Running the code

For running the code we first have to activate the Tensorflow environment. Once that is done, we access the folder where the skeleton of the code is saved, and type the following line in the command window:

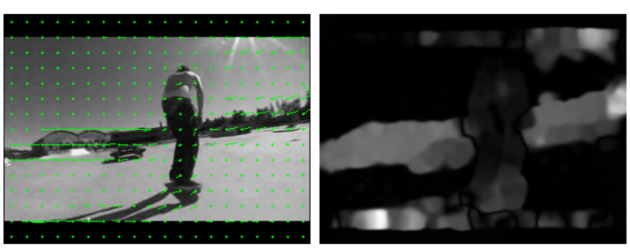
python lab1_skeleton.py videofile.format deltaT

Where *videofile.format* is the video sequence with which we want to work, and *deltaT* is the distance between computed frames, which should be an integer number.

IV. RESULTS AND ANALYSIS

A. deltaT parameter

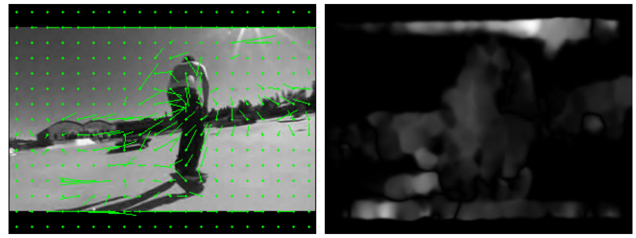
This parameter determines the distance between the two frames in which we are going to compute all the functions mentioned in III. The higher this value is, the more distance between frames, therefore the higher the error will be. In order to measure this how the results are different depending on the *deltaT* value, we have made two tests with the video sequence *v_SkateBoarding_g25_c01.avi* with values 3 and 25 for *deltaT*.



(a) Flow (b) GME Error

Fig. 1: Frame 150 with $deltaT = 3$

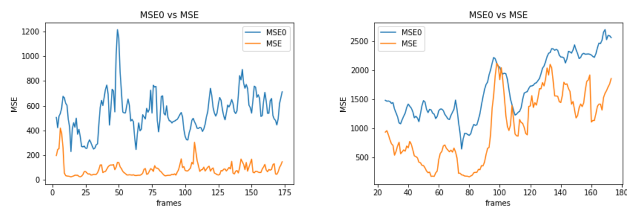
As we can see in figures 1 and 2, there is a huge difference of behaviour depending on the distance between frames. In the case of 2a the vectors are much longer and disperse than in the case of 1a, because the change between



(a) Flow (b) GME Error

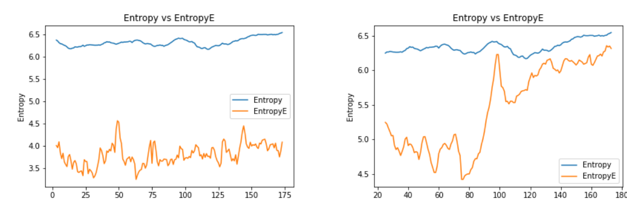
Fig. 2: Frame 150 with $deltaT = 25$

frames with distance 25 is much bigger than between frames of distance 3. This also affects the GME error, as can be seen comparing images 1b and 2b, where in the case of *deltaT* of value 3 the shape of the GME error is more defined and less caothic than in the other case.



(a) MSE with $deltaT = 3$ (b) MSE with $deltaT = 25$

Fig. 3: Mean Squared Error graphs



(a) Entropy with $deltaT = 3$ (b) Entropy with $deltaT = 25$

Fig. 4: Entropy graphs

From figure 3 we can see how the mean square error is not reduced very much when calculating the compensated frame if we use a high value of *deltaT*, but it is very well reduced for close frames. Also, overall values are much higher in the first case.
In the case of entropy, that we can see in figure 4, we see how the entropy error is high in the case of a high value of *deltaT*, but way smaller if this value is small.
For these reasons, we have decided to use small values of *deltaT* from now on.

B. PersonConvergence_720.MOV

The first video sequence used for the purpose of analysing is the Personconvergence_720.MOV video sequence. In this video sequence, the study of behaviour done in two different frames for the purpose of visualization: one in which the person moving and there is small movement of bike (Figure 5) and another the person is moving away from the camera (converging towards the center) and there is also bike and car movement (Figure 6) the study has been done using delatT 5.

As we can observe form (Fig. 5b) the motion vector are very short and in different direction and concentrated around the moving person also on the moving bike. However in (Fig. 5c) since Global motion estimation are applied the motion vectors they are on the same direction. The difference between GME and Flow depicted in (Figure 5d) shows one big blob in the middle of the frame which indicates the person and a very small white blob which indicates moving bike. This error has occurred due to motion vectors that are in opposite direction from the global motion of the object.

As we can infer from (Figure 5f) and (Figure 6f) are way smoother than (Figure 5e) and (Figure 6e), respectively, due to the performed compensation in the frames, which means that the error is reduced. This is also related to the entropy error measure, shown in image 7b.

We also observed that as the person moves far away from the camera(towards the center of the frame) the motion vector will become mostly in the same direction as the object is only seen from behind and getting faraway. the GME for this video sequence looks good as we can see the GME Error (distribution of the energy of residual motion) form Figure 5d and Figure 6d the sky is clear and where as for the moving object we can see that there are some blobs that correspond to the person and the moving object.

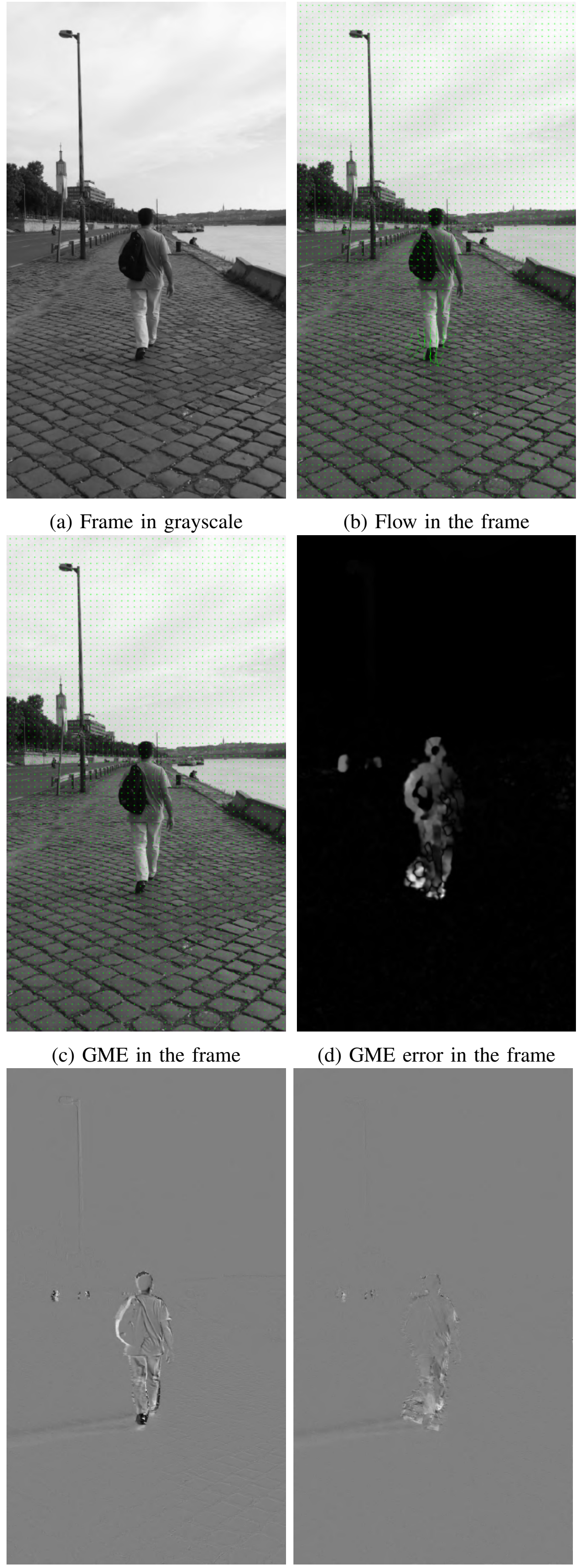
When compensating the frames, we can see that the MSE is reduced notably (Figure 7a), and in consequence the PSNR (Figure 7c) is increased, as they are inversely related. Due to compensation, the entropy error (Figure 7b) is also small, producing an improvement in the result.

C. *Birds_720.MOV*

The second video sequence analyzed has been the *Birds_720.MOV* video sequence. This video is interesting due to the shifting of the camera during the recording, which provides some difference from the video used in section IV-B. The study has been made with a *deltaT* value of 3.

We will study the behaviour of the different parameters in two different frames: one in which the birds haven't started to fly yet (Figure 8) and another in which there are birds flying already (Figure 9).

While in the image 8b there are small vectors, all of them horizontal due to the camera movement, in the image 9b we can see the vectors that are generated because of the movement of the water, the trees in the background, and specially from a bird that is flying.
In the case of the images 8c and 9c there are small horizontal vectors that show how the camera is rotating in the X axis from one frame to the other, but larger arrows in the second case as the rotation is bigger. The GME difference can be seen in images 8d and 9d. In the first case we see an image with big black zones and only small grey blobs that represent



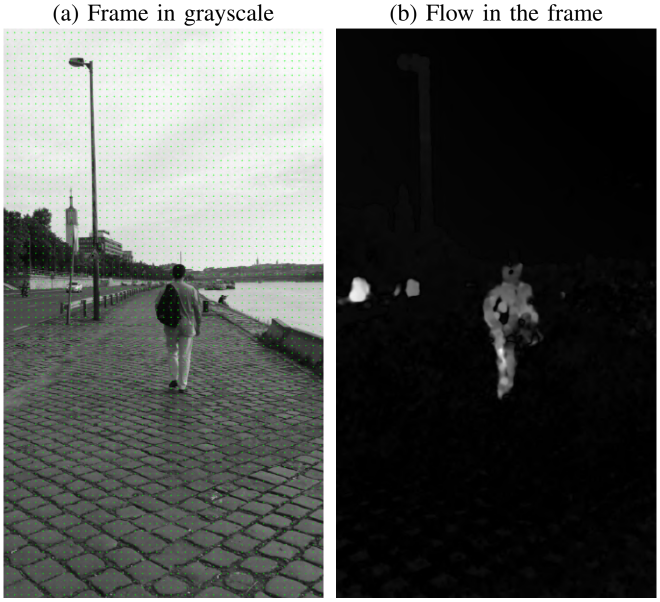
(a) Frame in grayscale (b) Flow in the frame

(c) GME in the frame (d) GME error in the frame

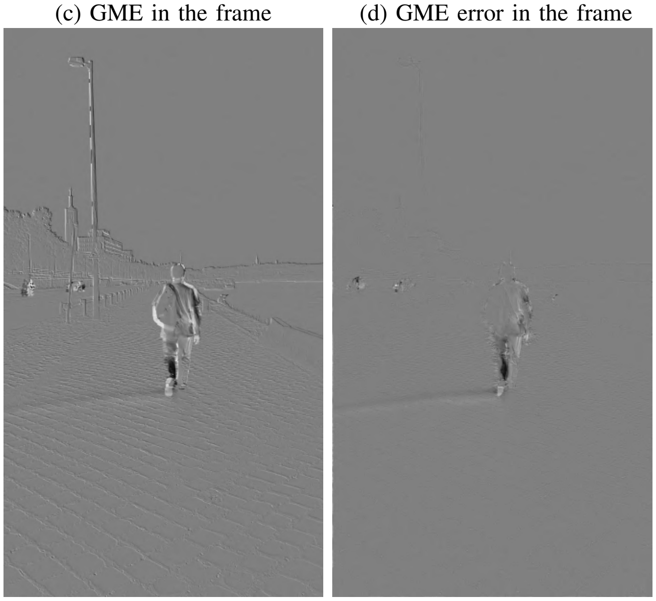
(e) Error without compensation (f) Error with compensation

Fig. 5: 20th frame of the PersonConvergence sequence

(a) Frame in grayscale (b) Flow in the frame

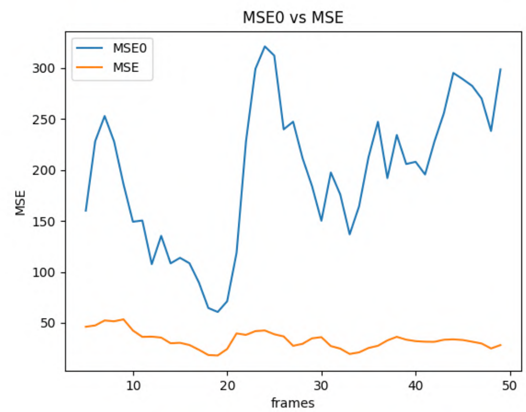


(c) GME in the frame (d) GME error in the frame

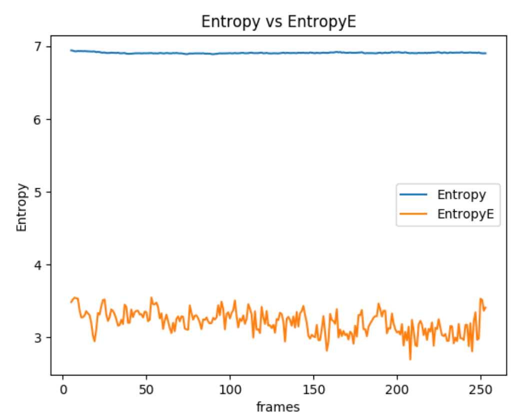


(e) Error without compensation (f) Error with compensation

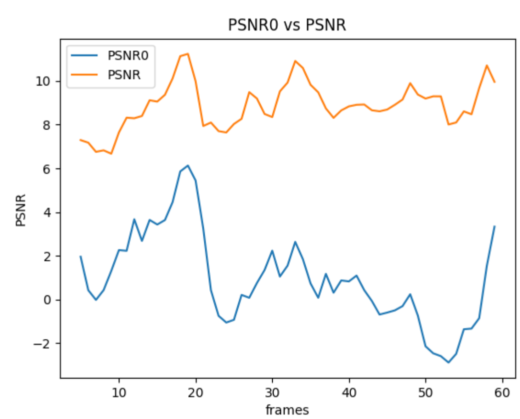
Fig. 6: 60th frame of the PersonConvergence sequence



(a) MSE graph

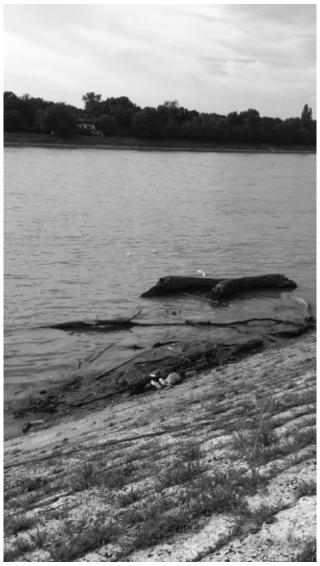


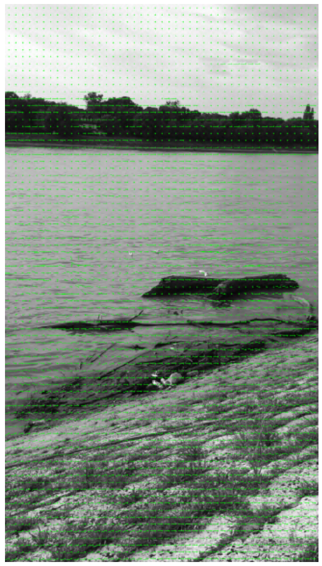
(b) Entropy graph

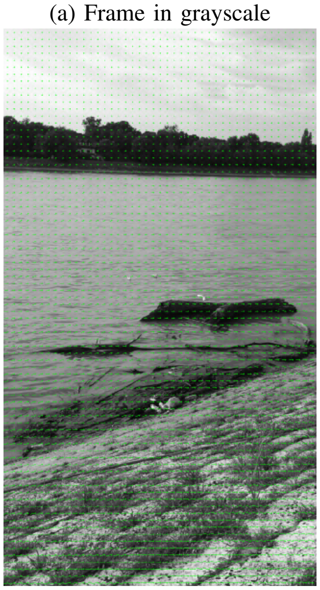


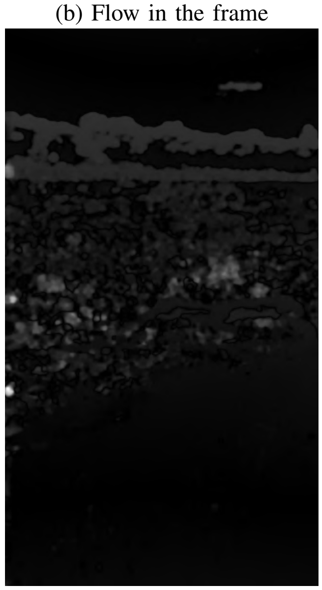
(c) PSNR_graph

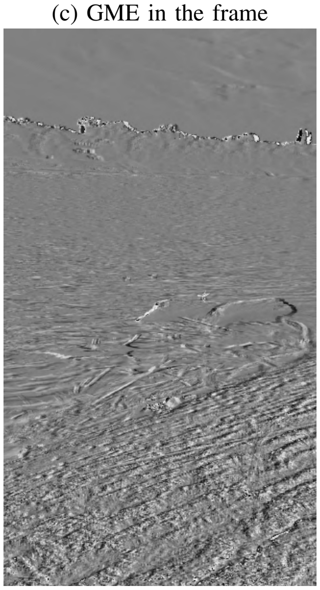
Fig. 7: Graphs of the PersoneConvergence sequence


(a) Frame in grayscale


(b) Flow in the frame


(c) GME in the frame


(d) GME error in the frame


(e) Error without compensation

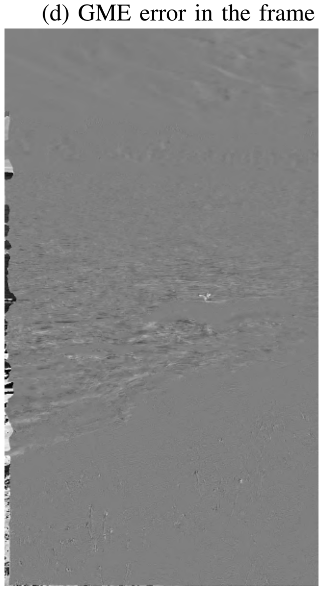
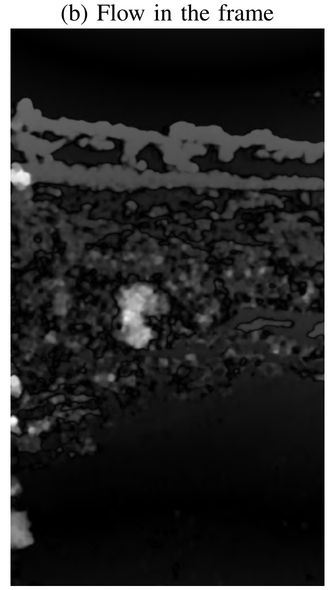

(f) Error with compensation

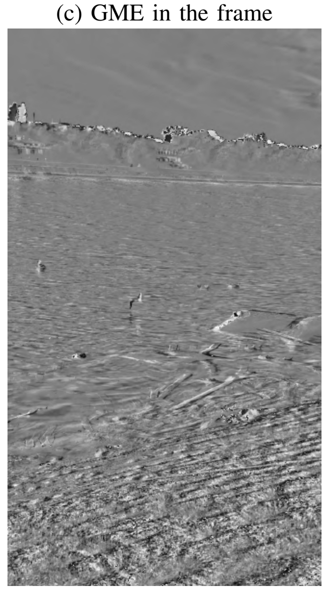
Fig. 8: 30th frame of the video sequence


(a) Frame in grayscale


(b) Flow in the frame


(c) GME in the frame


(d) GME error in the frame


(e) Error without compensation

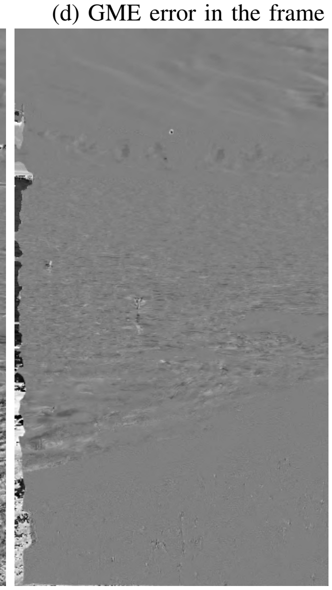

(f) Error with compensation

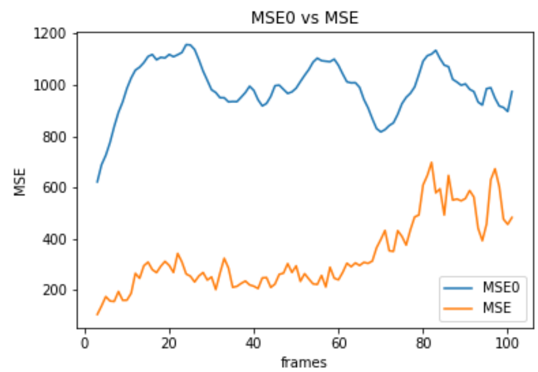
Fig. 9: 75th frame of the video sequence

the water movement, while in the second one there is a big blob where the bird is located (as it is the point in which a bigger movement has been performed). There are also some gray blobs that represent the movement of the water, and then the area of the sand and the sky, as the shifting is minimal, is represented by a very dark region.
We can also see that images 8f and 9f are way smoother than 8e and 9e, respectively, due to the performed compensation in the frames, which means that the error is reduced. This is also related to the entropy error measure, shown in image 10b.

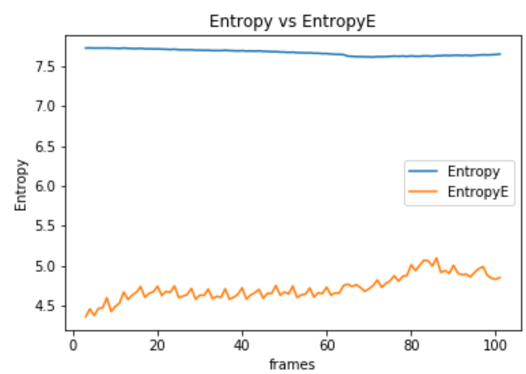
When compensating the frames, we can see that the MSE is reduced notably (image 10a), and in consequence the PSNR (image 10c) is increased, as they are inversely related. Due to compensation, the entropy error (image 10b) is also small, producing an improvement in the result.
In the graphs of images 10a and 10b we can see that the peaks, which correspond to more error and more entropy, are related to the frames in which the activity is higher, in this case, when more birds are flying, especially the one closer to the camera.

V. CONCLUSIONS

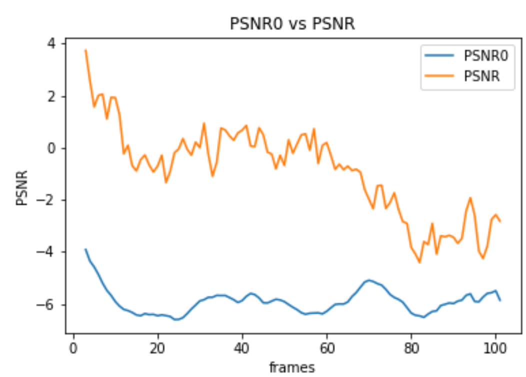
We can conclude that the smaller the initial frame difference we select with the *deltaT* value, the better and more accurate the result is, as it is not working with movements performed long time ago.
We have also conclude that, for a proper error calculation, the values should be absolute and normalized, otherwise the image will contain numerous errors and noise.
The use of compensated frames help to calculate properly the flow of the image, and avoid big noise and errors. We can say that Global motion estimation is a good motion estimation technique as it can give us the general(global) motion of the object.



(a) MSE graph



(b) Entropy graph



(c) PSNR graph

Fig. 10: Graphs of the video sequence