Program to estimate motion vectors of an image sequence using exhaustive search block matching algorithm.

Author: Siffi Singh

Student ID: 0660819

Dated: 1/12/2017

The files attached contains the MATLAB code for computing the motion vectors, entropy values, MSE and frame difference entropy.

-- --

Here in this report, I will be explaining:

I. Steps to run this file

II. Explanation of code logic and results

III. Inferences from result

IV. Conclusion

-- --

I. Steps to run the file:

-- --

1. Keep all the sequence of Claire.raw and Flower.raw files in the same folder as the MATLAB files.

2. There are six matlab files submitted:

- main.m
- Read_raw.m
- motionEstEs.m
- costFuncMAD.m
- minCost.m
- motioncomp.m

Enter the name of the pair of the file you want to see the motion vectors, and calculate the other parameters for.

3. So just two steps to run the file:

- **Enter the names of pair of files in main.m that you want to execute. Eg. Claire1.raw & claire2.raw**
- **Run main.m and you are done.**

4. Immediately after you run you will see the images you have inputted printed in Figure1, next you'll see the image of motion vectors in Figure2, next is the image after taking the frame difference in Figure3, Figure 4 contains the Motion compensated image.

-- --

II. Explanation of code logic and results

-- --

PART – 1

AIM: Calculate motion vectors using exhaustive search algorithm

1. What is exhaustive search algorithm?

2. Exhaustive Search Algorithm is a block matching algorithm, a way of locating matching macroblocks in a sequence of digital video frames for motion estimation.

3. Steps taken are:
First, we take Image1, and store it a 2D array, similarly we take the second image and store it in an 2D array. In the program above, I have used read_raw() function to read .raw files from the folder.

4. Next we pass the images to the function motionEstEs(), this function is the base function containing the main algorithm, i.e the implementation of the exhaustive search algorithm.

5. The block size mentioned in the program is 16, we divide our image into blocks of 16*16, that means:

- For sequence of Claire image, the original size if 288*352, so the no. of blocks in the image is 18*22.
- For sequence of Flower image, the original size if 480*720, so the no. of blocks in the image is 30*45.

6. Next we move one by one, taking each block and comparing it with pels given 7, i.e. the reference block would be fixed in Image 1, while we move from -7 to +7, both in horizontal direction in Image2, and look for closest match of frame.

7. We use the MAD(Mean Absolute Difference) as a measure, which has been calculated in costFuncMAD() function. For each reference frame, there will be [-7,7], horizontal and vertical, that is 15*15, different frame MAD values.

8. From the computed MAD values, we take the minimum cost from the minCost() function, and repeat this process for each reference block in Image1, until all the blocks have been computed.

9. The function motionComp is for computing the motion compensated image, so that we can compute the entropy values for MCFD in Part 4.

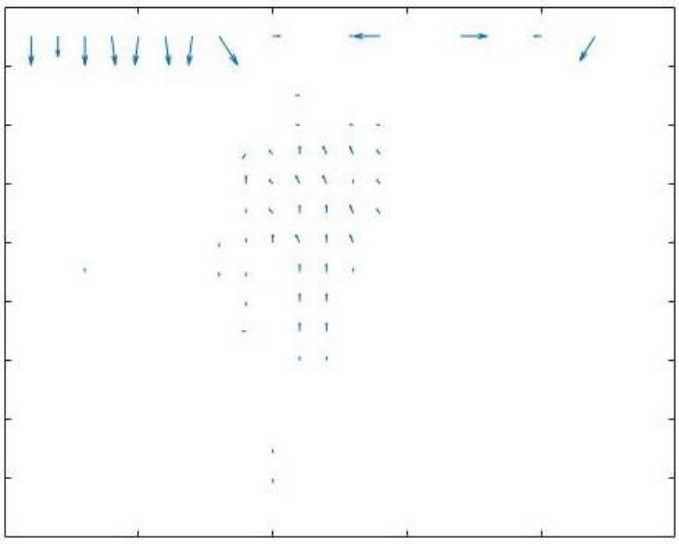
10. The motion compensated image generated in motionComp can be used to calculate entropy values and for comparision with image2 to check difference with the original image.

Hence, the algorithm is pretty smooth, it’s easy to run, just run the main function, and in the command window you can get the entropy values for the horizontal and vertical component and also you will get the MSE for the FD, also, the entropy of FD, the MSE for the MCFD and the entropy of the MCFD.

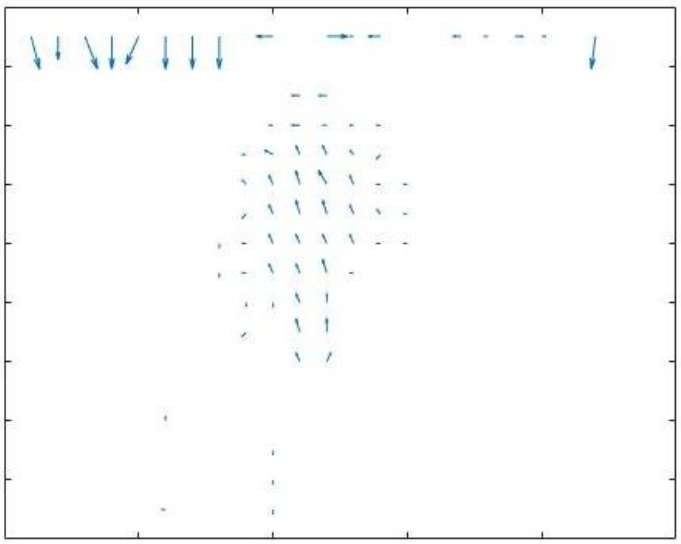
RESULTS -- || CLAIRE ||

-- --

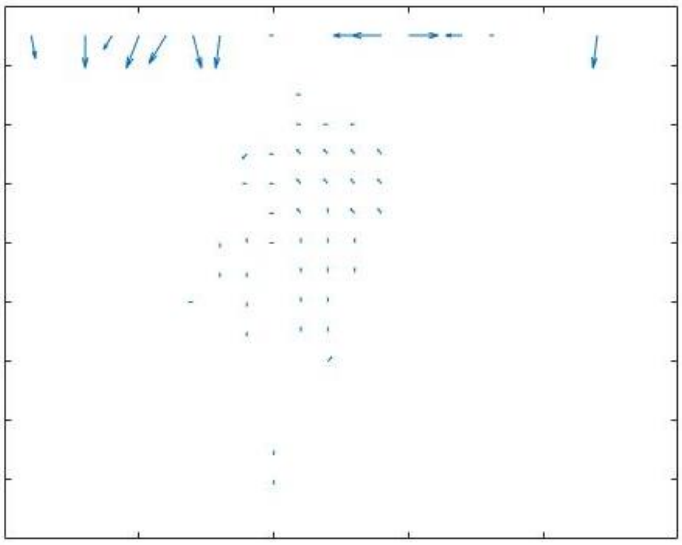
Claire1 and Claire2



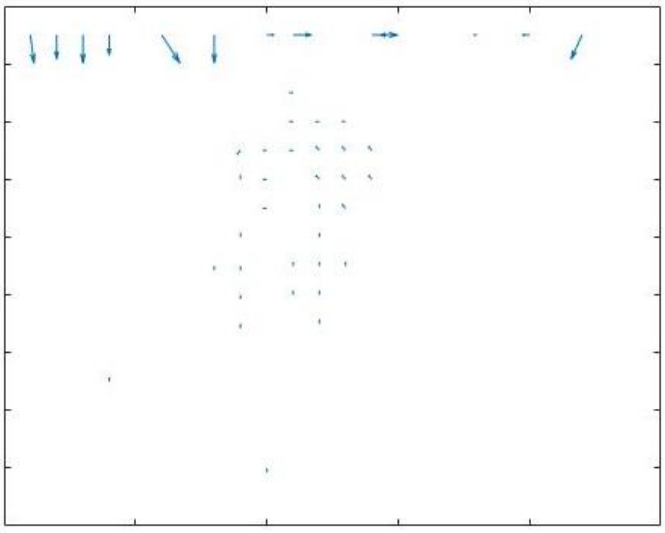
Claire2 and Claire3



Claire3 and Claire4



Claire4 and Claire5

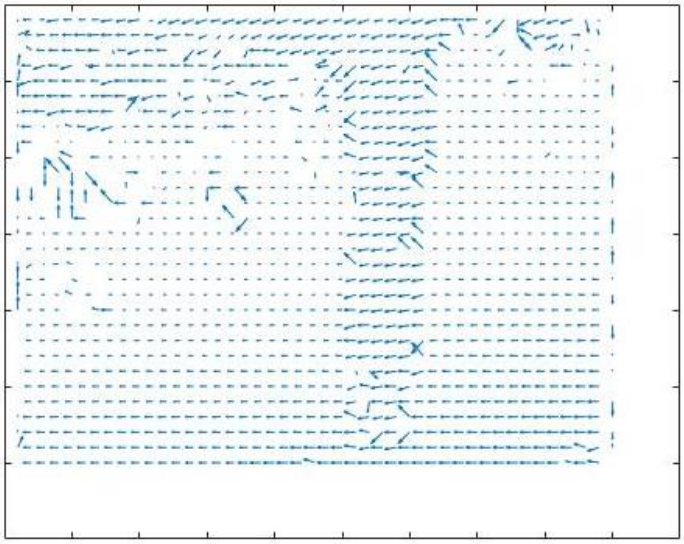


Note: The motion vectors as we move to different sequences of images of Claire, we see that the arrows are pointing upwards and from the original image also we see that the head of Claire is moving upwards.

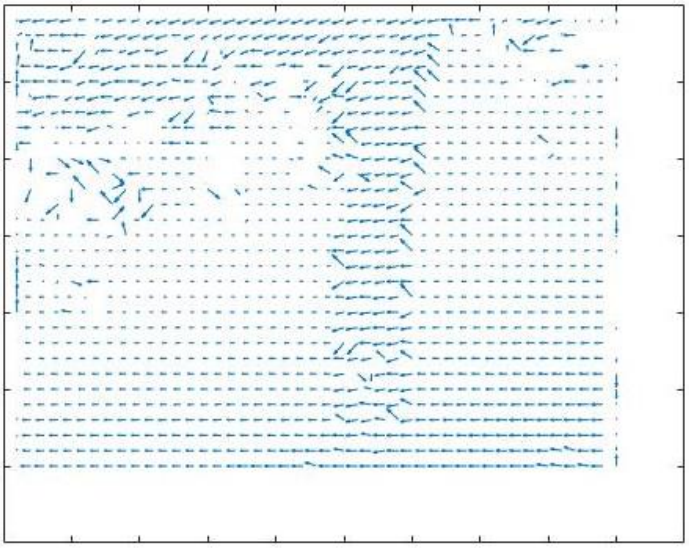
RESULTS -- || FLOWER ||

-- --

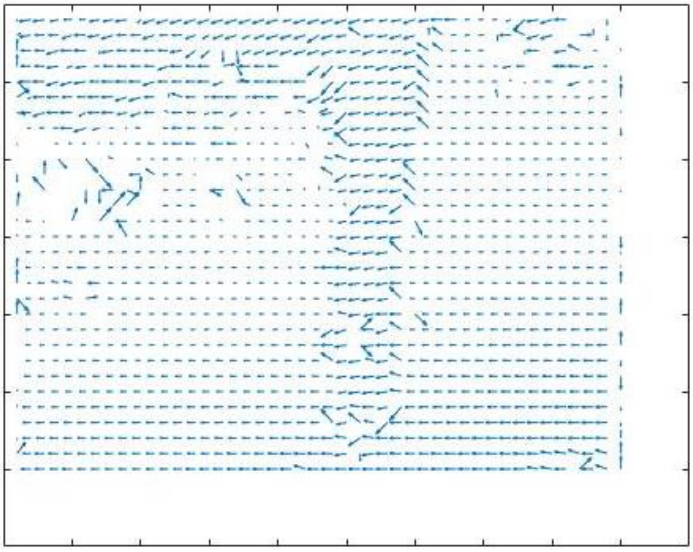
Flower1 and Flower2



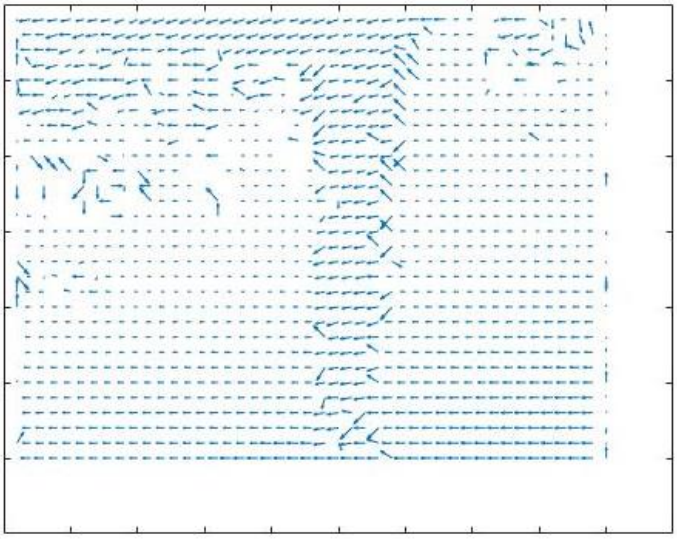
Flower2 and Flower3



Flower3 and Flower4



Flower4 and Flower5



Note: The motion vectors as we move to different sequences of images of Flowers, we see that the arrows are pointing left and from the original image also we see that the tree and the house is moving left, and it's pretty fast so there a lot of motion vectors.

PART – 2

AIM: Calculate motion vector entropies of horizontal and vertical motion components of motion vectors.

1. Entropy calculation is just taking the frequencies values from the horizontal and vertical component of the motion vector of each pair of image sequences of Claire and flower and then taking the average for each set of image.

2. Calculating the entropy values of summation(–p*log(p)) where p: probability values.

3. The process is to take the X-coordinate values of the motion vectors for the horizontal component and Y-coordinate values of the motion vectors for the vertical component and calculate the probability of each value to calculate entropy as per above formula.

RESULTS

-- --

CLAIRE		
	Entropy (Horizontal)	**Entropy(Vertical)**
Claire1/Claire2	0.244142	0.09774
Claire2/Claire3	0.277752	0.113274
Claire3/Claire4	0.244142	0.09774
Claire4/Claire5	0.208383	0.09774
Average	**0.24360475**	**0.1016235**

Table 1: Entropy horizontal and Entropy vertical values for sequence of images of Claire

FLOWER		
	Entropy (Horizontal)	**Entropy(Vertical)**
Flower1/Flower2	0.69129	0.157768
Flower2/Flower3	0.730728	0.14558
Flower3/Flower4	0.69129	0.17353
Flower4/Flower5	0.727815	0.128775
Average	**0.71028075**	**0.15141325**

Table 2: Entropy horizontal and Entropy vertical values for sequence of images of Flower

PART – 3

AIM: Calculate the mean square error of (a) Frame differences (b) Motion Compensated Frame Differences of each picture pair.

1. Frame Difference is the difference for the pixel values between Image1 and Image2.

2. Motion Compensated Frame is the frame after moving the Image1 pixels with respect to motion vectors calculated, and then comparing it with Image2, the actual image.

3. The mean square error has been calculated the usual way, taking the absolute difference, squaring it and taking the mean value.

CLAIRE		
	MSE_FD	MSE_MCFD
Claire1/Claire2	15.461283	4.238586
Claire2/Claire3	30.152669	7.128807
Claire3/Claire4	9.679944	3.06089
Claire4/Claire5	6.549233	2.409075
Average	**15.46078225**	**4.2093395**

Table 3: MSE of FD and MSE of MCFD values for sequence of images of Claire

FLOWER		
	MSE_FD	MSE_MCFD
Flower1/Flower2	1483.652115	212.674413
Flower2/Flower3	1519.318061	215.518944
Flower3/Flower4	1587.793996	222.052472
Flower4/Flower5	1565.024795	219.859042
Average	**1538.947242**	**217.5262178**

Table 4: MSE of FD and MSE of MCFD values for sequence of images of Flower

RESULTS (FRAME DIFFERENCE) - || CLAIRE ||

-- --

FD Claire1 and Claire2

FD Claire2 and Claire3

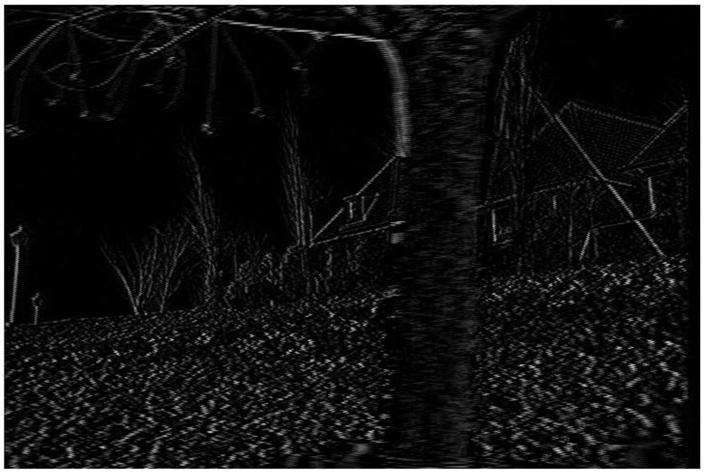
FD Claire3 and Claire4

FD Claire4 and Claire5

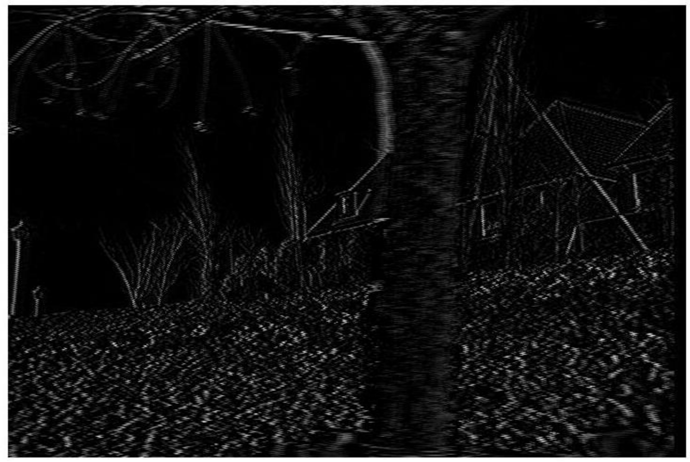
RESULTS (FRAME DIFFERENCE) - || FLOWER ||

-- --

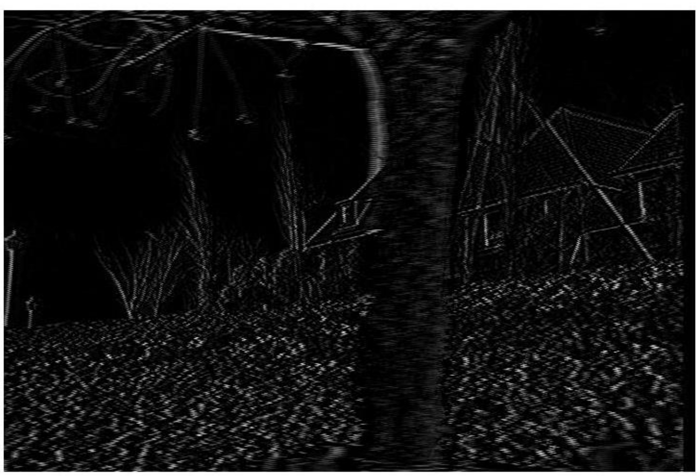
FD Flower1 and Flower2



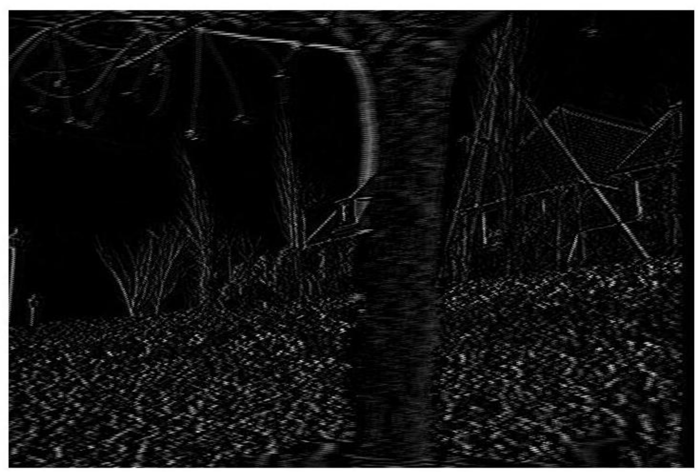
FD Flower2 and Flower3



FD Flower3 and Flower4



FD Flower4 and Flower5



RESULTS (MOTION COMPENSATED FRAME) - || CLAIRE ||

-- --

MCFD Claire1 and Claire2

MCFD Claire2 and Claire3

MCFD Claire3 and Claire4

MCFD Claire4 and Claire5

RESULTS (MOTION COMPENSATED FRAME) - || FLOWER ||

-- --

MCFD Flower1 and Flower2

MCFD Flower2 and Flower3

MCFD Flower3 and Flower4

MCFD Flower4 and Flower5

PART – 4

AIM: Calculate the average entropy of the FD and the MCFD obtained for each sequence separately.

1. Computing average entropy of the FD, i.e. for 0-255 values, calculating frequency and the probabilities, and future entropy values for each pair.

2. Finally we will take the average value of entropies for each sequence of images.

RESULTS

-- --

CLAIRE		
	Entropy _ FD	**Entropy_MCFD**
Claire1/Claire2	1.508832	1.590651
Claire2/Claire3	1.396277	1.474521
Claire3/Claire4	0.879528	0.963359
Claire4/Claire5	1.376049	1.465165
Average	**1.2901715**	**1.373424**

Table 5: Entropy FD and Entropy MCFD values for sequence of images of Claire

FLOWER		
	Entropy _ FD	**Entropy_MCFD**
Flower1/Flower2	3.711712	3.924461
Flower2/Flower3	3.730971	3.930553
Flower3/Flower4	3.755494	3.941105
Flower4/Flower5	3.751291	3.937379
Average	**3.737367**	**3.9333745**

Table 6: Entropy FD and Entropy MCFD values for sequence of images of Flower

-- --

III. Inferences from result:

-- --

1. The result shows that there are fewer vectors in Claire and more no. of vectors in Flower which makes sense because the sequence of images of Flower is moving faster than the sequences of image of Claire.

2. The value of MSE for Frame Difference is greater than the MSE of the entropy of MCFD, and this is because the image, Motion compensated frame, the predicted image is closer to the second image and thus the difference between the two should be less than the difference between the image 1 and image 2.

3. The value of entropy for Frame Difference and the entropy value of MCFD also shows that because the Motion compensated frame is more close to the second image considered for computation, hence we entropy values decreases for MCFD.

4. In all, motion estimation using block searching exhaustive search algorithm helps us to predict the next image in the sequence using motion vectors and thus contributes in doing data compression.

-- --

IV. Conclusion:

-- --

1. This algorithm calculates the cost function at each possible location in the search window.

2. This leads to the best possible match of the macro-block in the reference frame with a block in another frame.

3. The resulting motion compensated image has highest peak signal-to-noise ratio as compared to any other block matching algorithm. However, this is the most computationally extensive block matching algorithm among all.

4. A larger search window requires greater number of computations.