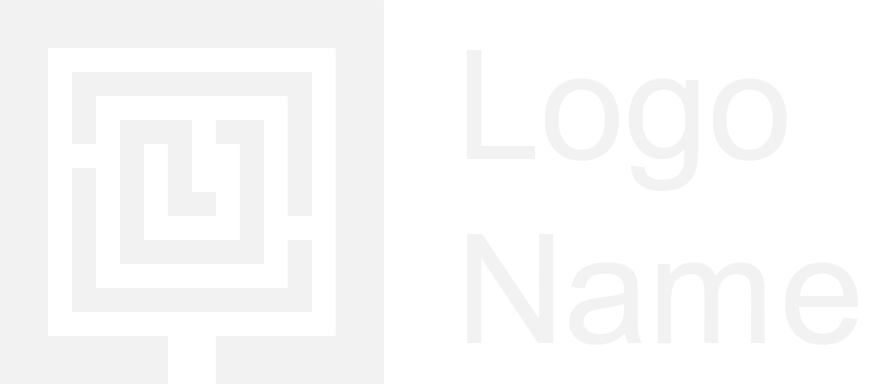


|  |
| --- |
| Digital Audio and video |
|  |
| December 29  Live Stream  Authored by: Fady sameh  Nadeem Mohamed  Basel Mohamed  Ahmed Salah |



Steps to run

There are 8 files used for the Project:

1. app2.mlapp
2. decoding.m
3. encoding.m
4. minCost.m
5. motionComp.m
6. motionEstEs.m
7. read\_raw.m
8. readbin.m
9. savebin.m

there are three steps to run the file:

1. Choose DCT (16x16 or 8x8 or 4x4)
2. Start Cam
3. Browse to Compression Folder

Immediately After you run you will see compressed image in image plane.

Explanation of code logic and results

AIM: Calculate motion vectors using exhaustive search algorithm

1. What is exhaustive search algorithm?

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

2.Steps taken:

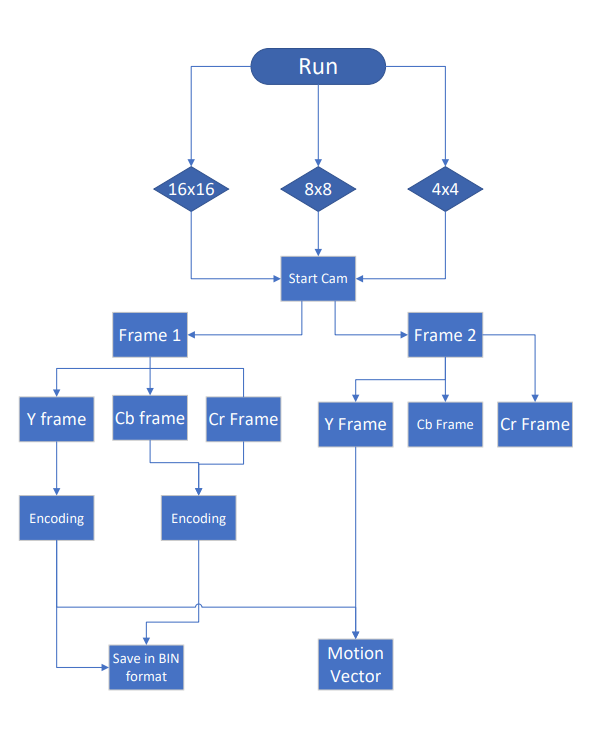
* First we choose DCT type.
* Second we press start cam:
* First frame divide to YCBCR format and decoded
* Second frame divide to YCBCR format
* Then we use function MotionEstEs() , this function is the base function containing the main algorithm, i.e the implementation of the exhaustive search algorithm, that takes Y from First frame and Y from Second frame with macroblock.
* The block Size in the program is 8 , we divide image into blocks of 8x8

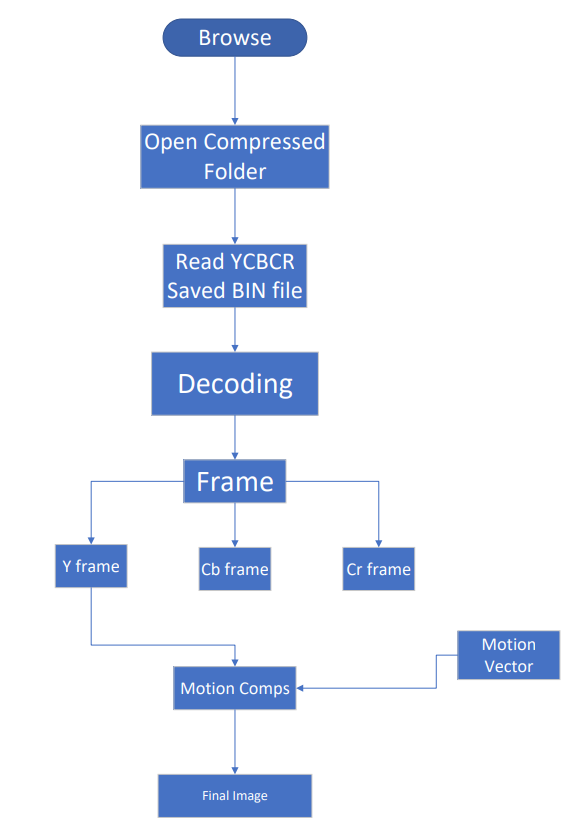
3. Next we move one by one, taking each block and comparing it with pels given 3, i.e. the reference block would be fixed in first frame, while we move from -3 to +3, both in horizontal direction in Second frame, and look for closest match of frame.

4. We use the MAD(Mean Absolute Difference) as a measure, which has been calculated in costFuncMAD() function. For each reference frame, there will be [-3,3], horizontal and vertical, that is 7\*7, different frame MAD values.

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

Flowchart





Comparison

16x16:

Original Image:



Final Image:



8x8:

Original image:



Final Image:



4x4:

Original Image:



Final Image:



Team Task:

Fady sameh, Ahmed Salah, Basel , Nadeem: Code

Ahmed Salah: Design

Basel, Nadeem: Documentation