Lab 12 Report Shuchen Zhang (szhan114)

**Introduction:**

In this lab, we implement the block matching method for motion estimation, which is frequently used in H.263 and MPEG 1-2. We apply block matching to a two-frame sequence and to find the motion vector within a certain search window that achieves the least mean square error between the prediction and the real image.

**Methods:**

In both part A and part B, we first partition the current frame into 16x16 non-overlapping subblocks and we only focus on the subblock that contains the black square object. Given the search range 63 which means the motion vector’s x and y values should be within [-31,31]. Then we examine the previous frame and compare the difference between each Previous (location – MV(location)) with Current(location) and calculate the mean square error. The motion vector that achieve the lowest mean square error would be our predicted motion vector.

A close up of a map

Description generated with high confidenceFor part A, the minimum mean square error is achieved by the motion vector (17,-17) and the mean square error = 0. The plot of mean square error w.r.t the motion vector is:

A close up of a map

Description generated with high confidenceFor part B, the minimum mean square error is achieved by the motion vector (1,-27) and its mean square error is 67.54. The plot of mean square error w.r.t. the motion vectors is:

Q1. The total number of operations per pixel for both Part A and Part B are the same since the block size and the search size are picked the same. There are 63\*63 possible motion vectors. For each motion vector, there is 16\*16 difference finding and adding to the square error sum. For each pixel within the block, there is 1 subtraction and 1 multiplication and 1 summation operation. There is 1 division by block size operation for each block. So in total there are 63\*63\*(16\*16\*3+1) = 3052161 pixel operations. The average pixel operation = total/(search size \* search size) = 769 operations per pixel.

Q2. If we increase the search size by a certain factor, while keeping the block size unchanged. The computation time will increase by that factor, but the prediction will be more accurate, since we are comparing more possible mean square errors to find more accurate motion vectors. If we increase the block size by a certain factor while keeping the search size the same, the computation time will approximately increase by that factor but will find a more accurate motion vectors as well. However, the block size cannot be too large, the target object will be very small compared to the block size and the match may not be meaningful. The block size cannot be too small, since it would cause more false matches due to the quantization noises.

Q3. The energy of the block used for block matching in Part B is 1.0303 e+7. This is calculated by summing the square of all the pixel intensity of the current frame block.

Q4. a. plot the histogram of the prediction error (absolute value of the difference) which is also the displaced frame difference for the block:

A close up of a logo

Description generated with very high confidence

b. the entropy of this histogram is : -6.59175, which is calculated according to the entropy function:

c. print out the DFD image. The DFD image:

![A blurry photo

Description generated with very high confidence]()

**Conclusion:**

In this lab, we implement the block matching method for motion prediction. Two 2-frame image sequences are examined. In block matching, we want to find the motion vector that could minimize the mean square error between the prediction block and the target block. By increasing the search size, we could improve the accuracy but may have coding overhead. Block size couldn’t be too large or too small. 16x16 is the standard size for block matching.