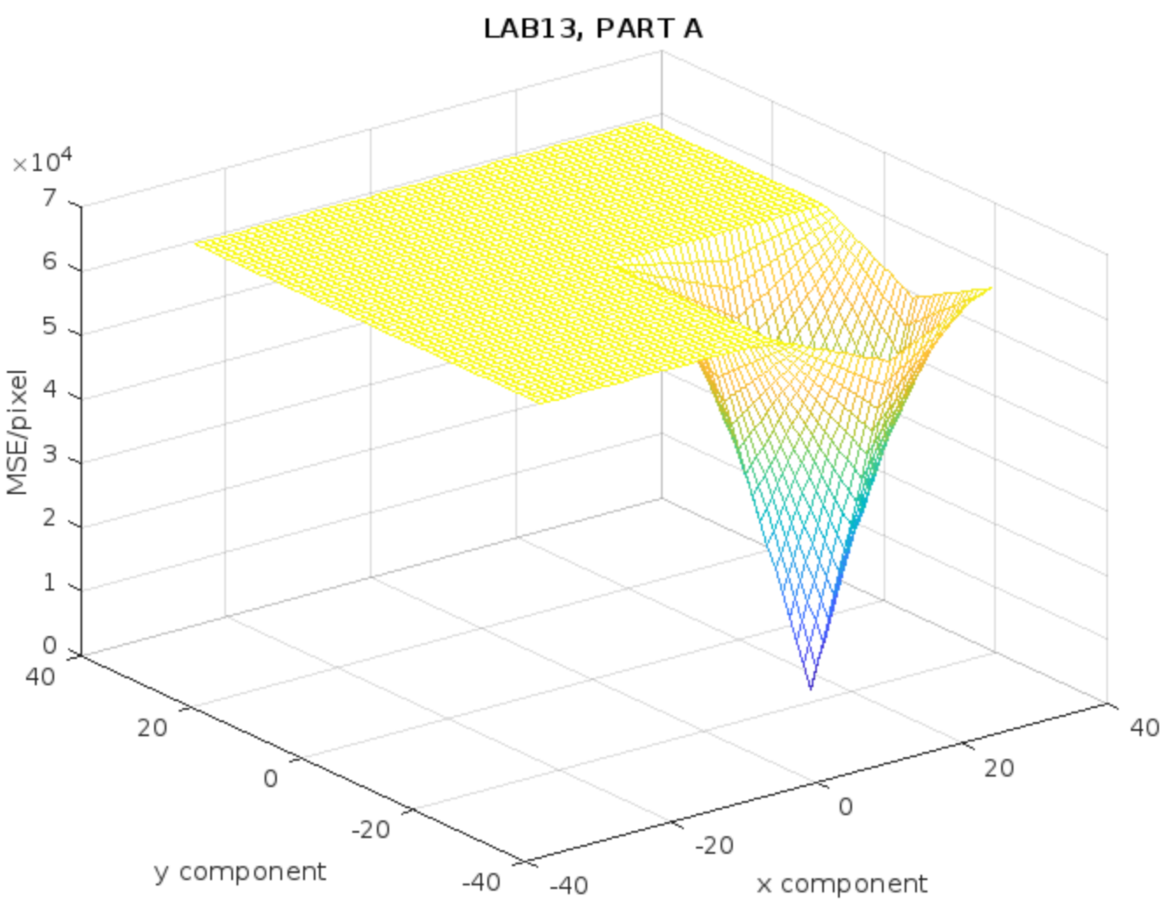
ECE 418 Lab 12

Aahan Thapliyal
Aahant2

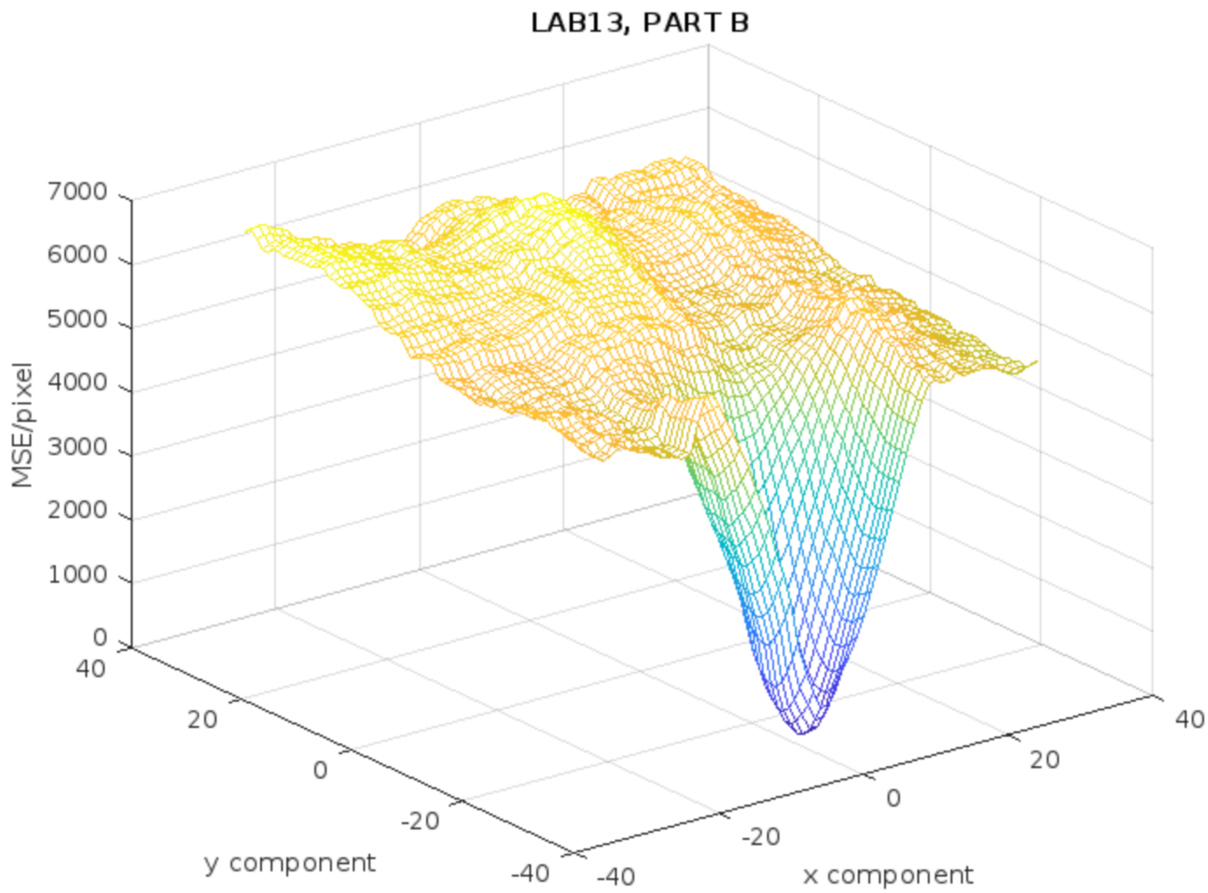
In this lab, we will explore the block matching method for motion estimation, which is commonly used in video compression standards such as H.263 and MPEG 1-2. We will apply this method to a two-frame sequence and attempt to find the motion vector that minimizes the mean square error between the predicted and actual images.

In both part A and part B, we begin by dividing the current frame into non-overlapping 16x16 subblocks. We then identify the subblock that contains the black square object and set a search range of 63, which restricts the motion vector's x and y values to the range of [-31, 31]. Next, we examine the previous frame and compare the difference between each pixel's previous location (calculated using the motion vector) and its current location. We calculate the mean square error for each subblock within the search range. The motion vector that results in the lowest mean square error is selected as our predicted motion vector.

For Part A, the minimum squared mean is achieved by the vector (17, -17) and as seen in the figure below the mse = 0. Plot of the mse w.r.t the motion vector is:



For Part B, the minimum squared mean error is achieved by the vector (1,-27) and its mse = 67.54. Plot of the mse w.r.t the motion vector is:



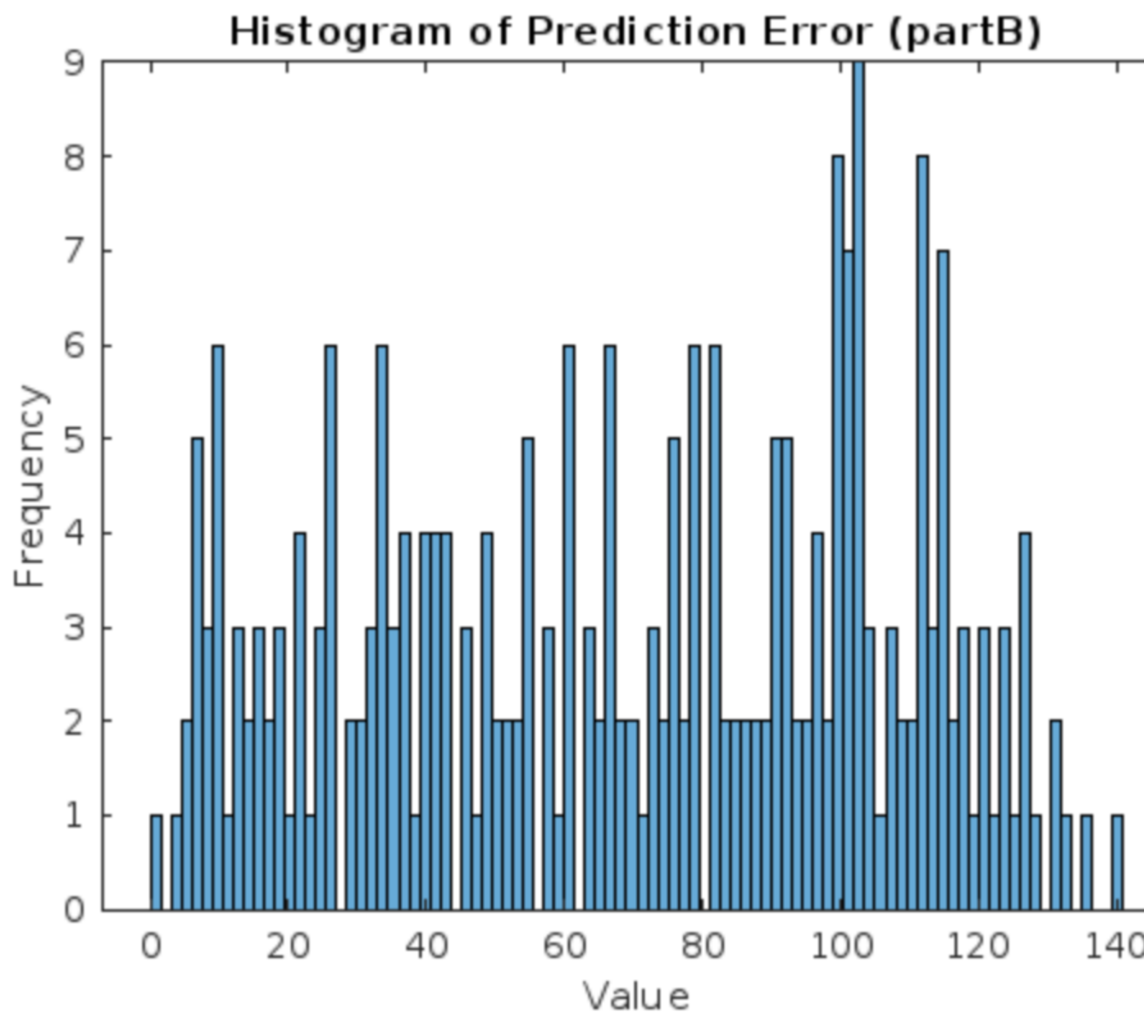
Q1. The total number of operations per pixel for both Part A and Part B is the same, since the block size and search size are identical. With a search range of 63, there are 63*63 possible motion vectors. For each motion vector, we calculate the difference between each pixel's previous and current location within the 16x16 block and add the resulting value to the square error sum. For each pixel within the block, we perform one subtraction, one multiplication, and one summation operation. We also perform one division by the block size operation for each block. Therefore, the total number of pixel operations is 6363(16163+1) = 3,052,161. The average number of pixel operations is calculated by dividing the total by the search size squared, resulting in an average of 769 operations per pixel.

Q2. Increasing the search size by a certain factor, while keeping the block size constant, will lead to a proportional increase in computation time. However, this increase will result in more accurate predictions, as it allows for a larger number of possible mean square errors to be compared and more accurate motion vectors to be identified. Similarly, increasing the block size by a certain factor while keeping the search size constant will lead to an approximately proportional increase in computation time, but also result in more accurate motion vectors being identified. However, the block size should not be too large, as the target object may be very small compared to the block size, making the match meaningless. Conversely, the block size should not be too small, as this can result in more false matches due to quantization noises.

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

Q4.

a) Histogram of Prediction Error:



b) Entropy: -6.592

c) DFD Image

In this lab, we are implementing the block matching method for motion prediction, using two 2-frame image sequences. The goal of block matching is to identify the motion vector that minimizes the mean square error between the prediction block and the target block. Increasing the search size can improve accuracy, but may also result in coding overhead. The block size should not be too large or too small, with a standard size of 16x16 typically used for block matching.