ELE725 LAB 4 (Motion Compensation and CBIR)

Name: Yun Peng Zou
FEAS
Toronto, Canada
ypzou@ryerson.ca

Abstract- In this lab, interframe frame encoding is explored. The algorithms tested were Motion Vector Computation and using the values calculated in DPCM the values were minimized. This lab also explores principles around content based matching in which the images is matched based on color histogram.

I. INTRODUCTION

In the modern era, image data has become very large as the resolution and bit per pixel goes up to create a better quality image. The size of videos goes up proportionally. To minimize the amount of bits that need to be sent, the difference between to frames are calculated which is inter-frame encoding. To find the best block from the image to encode, motion vector computation is used to minimize the number of bits.

II. THEORY

In inter-frame coding, the image is spilt into block and the frame is coded based on the previous frame. Rather than the image getting sent every frame, the difference frame is sent which is much smaller than the actual frame. However, to minimize the size even more, motion computation algorithm can be used. Using this algorithm rather than just using the previous value, the motion computation algorithm matches the block with a set of candidate blocks using a given radius. The equation is given below: [1]

$$MAD(i,j,p,q) = \frac{\sum_{p=1}^{m}\sum_{q=1}^{n} |C_{n+1}[p,q] - C_n[p+i,q+j]|}{mn} \qquad (1)$$

For content based retrieval, to find a matching image template. A histogram of the image is created. The image is matched by using one of these three similarity metrics:[1]

Manhattan (cityblock) distance:

$$d_{L1}(h,g) = \sum_i |h(i) - g(i)| \qquad (2)$$

Euclidean distance:

$$d_{L2}(h,g) = \sum_i (h(i) - g(i))^2 \qquad (3)$$

Histogram Intersection:

$$d_{int}(h,g) = \frac{\sum_i \min(h(i), g(i))}{\min(|h|, |g|)} \qquad (4)$$

III. METHODOLOGY

A. Part 1 Motion Vector Computation

The Video was read with the VideoReader function in MatLab. The image was converted to grayscale using rgb2gray(frame2). The size of the image was obtained using the size function which gives you x pixel by y pixel. To obtain the number of block per columns, the dimension of x pixel is divided by 16 and ceiling function is used and same for rows except the axis is the y axis. In a double for loop iterator though each block, call the function computeMotionVect with the two frames , x*16+1, y*16+1, radius of search area = 7 since the array starts at 0 and each block size is 16. For the computeMotionVector, the lowest difference value is preset to 999999. If the given row + radius or given column+ radius is larger than the dimensions of the image than return the original row, column since it means it will search out of bounds. In four for loops, x and y is used as the given row and given column using rx,ry as the search area radius in x and search area radius in y. The difference is got though the forumula diff =diff+ abs(frameCurr(x+rx,y+ry)- framePrev(x,y). If it is found that the new difference is lowered than the old difference, set new difference to old difference and set dx,dy to the new values. Every time rx, or ry increases set difference to 0 since it is a new search area. In a double for loop, concatenate the blocks together using the information from the from compute Motion Vector.

B. Part 2 Content Based

To read the image file use the imread to read the image file. To change the color space from RGB to HSV use the command rgb2hsv in matlab. To make a histogram of the image, use the build in matlab function histogram to compelete it.

IV. RESULTS

A. Part 1 Motion Vector Computation

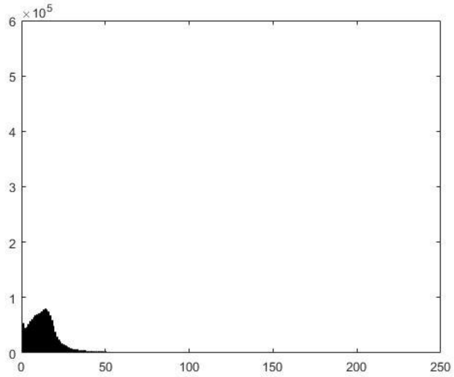

Figure 1: Histogram of Difference using previous frame

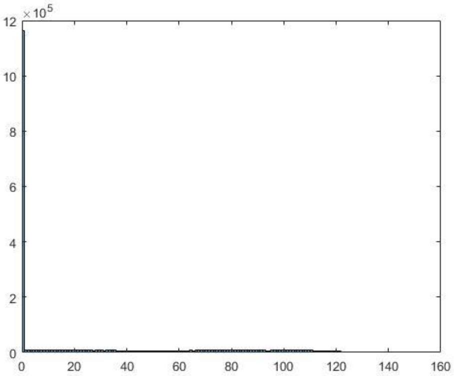

Figure 2: Histogram of Difference using motion vector computation

Entropy of difference using previous frame = 4.6443

Entropy of difference using motion vector = 4.0950

B. Part 2 Content Based

Picture Number	Difference In histogram
2	1.2569e+06
3	4.8338e+05
4	9.447e+05
5	5.068e+05

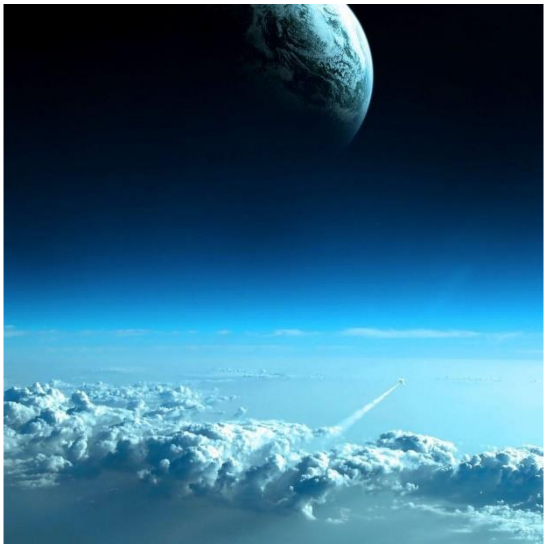

Figure 2: Original Image

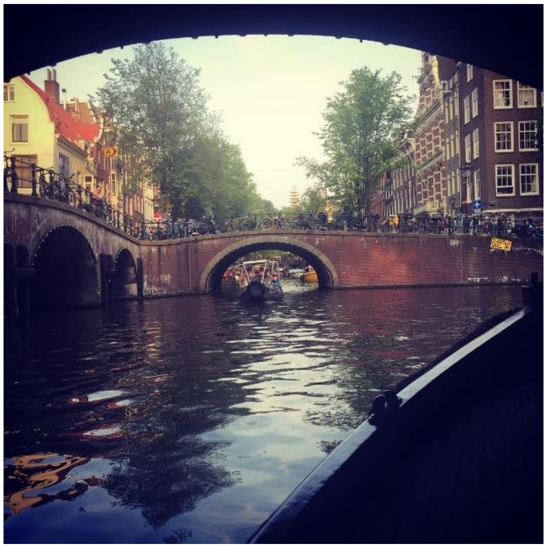

Figure 3: Lowest Difference

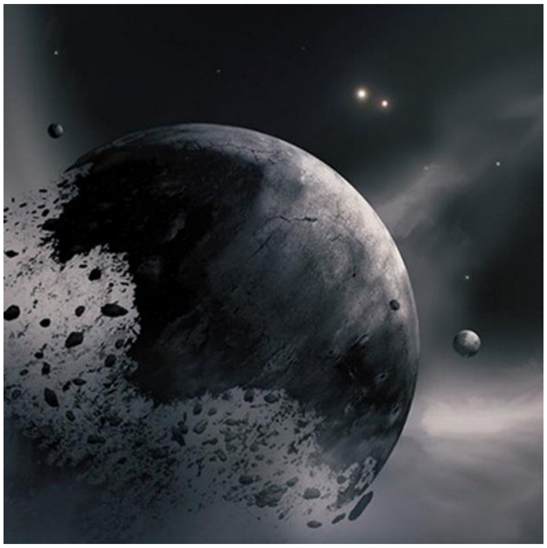

Figure 4: 2nd Lowest Difference

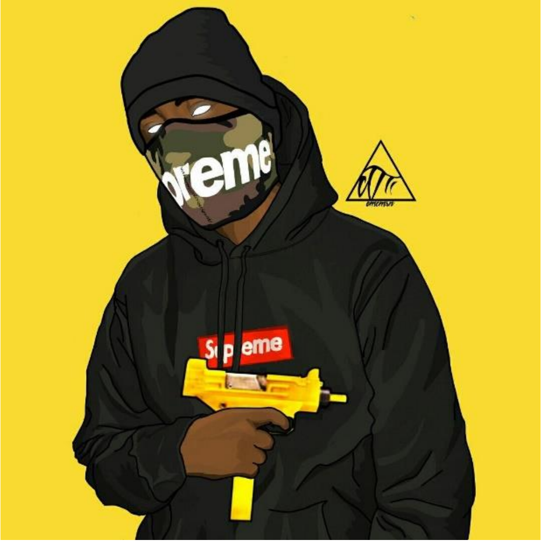

Figure 5: 3nd Lowest Difference

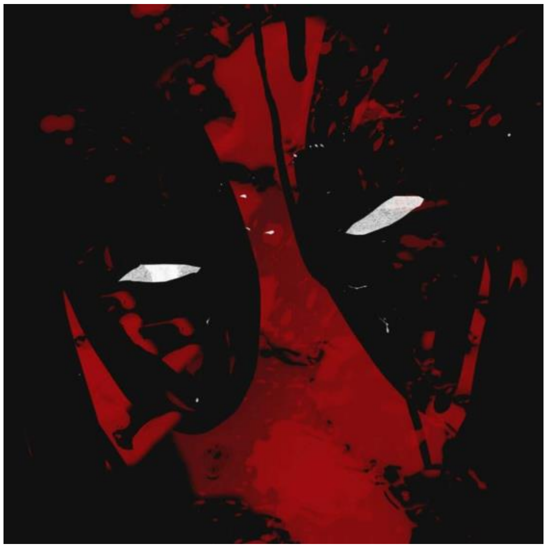

Figure 5: 3nd Lowest Difference

V. DISCUSSION

A. Part 1 Motion Vector Computation

Using the motion vector computation, the entropy value of the image goes down by 20 percent. From the histogram it is seen that the values that use to be concentrated between 0 and 25 became concentrated between 0 and 10 when comparing figure 1 and figure 2. It can be seen that the motion vector computation compresses the image by a sizeable chunk but the disadvantage is caculating the block. With higher compressing, the search radius will need to be increased which means the computation time increases as well.

B. Part 2 Content Based

Using the content based image dectection, it is seen that it matches images based upon how far the distance between two image color space as seen from the order from figure 3 to figure 5;

VI. CONCLUSION

In conclusion, motion vector computation further compresses the image size. The trade off the compression is the computation time which is related to the size of search zone. A balance between computation time and compression makes the algorithm effective. The content based analysis uses the color space to calculate difference.

REFERENCE

[1] Faculty of Engineering and Architectural Science. (2020, Winter). ELE725 Lab 4 Manual: Motion Compensation and CBIR. Ryerson University.
.