ELE725 Lab 4 Report

Motion Compensation and CBIR

Ashif Fahim
Department of Electrical, Computer and Biomedical Engineering
Ryerson University
Toronto, ON, Canada

Santos Bethany
Department of Electrical, Computer and Biomedical Engineering
Ryerson University
Toronto, ON, Canada

***Abstract*— Motion Vector Computation was used on two single frames from a video then performed DPCM where one of the frame have the motion vector shift applied to it. It was found that the entropy of the the frame difference with the motion vector shift was smaller compared to the one without. A small image database was tested using local and global histograms with various distance metrics for content based retrieval analysis. The conclusive evidence found, using local histograms gave more accurate r**

I. Introduction

In this lab, two alternative template matching algorithm will be implemented. The first method is the Motion Vector Computation considered in the Motion Compensation algorithm within a video code. The other will be an image matching exercise. For this, colour histograms will be used to index a small collection of images, then match a query image against the collection and order the collection in terms of degrees of similarity

II. Theory

A. Motion Vector Computation

One major cause for frame difference in a video sequence is due to motion. Motion can be from camera movement or movement of the objects in the video itself-this motion causes large inter frame differences which would result in a higher entropy value. To reduce the entropy, it can be assumed that there has been some motion in the video and groups of pixels have moved to a nearby region; this group of pixels may be a better representation for the predictor than just the pixel adjacent for applying DPCM. A visualization of this explanation is seen in Figure 1.

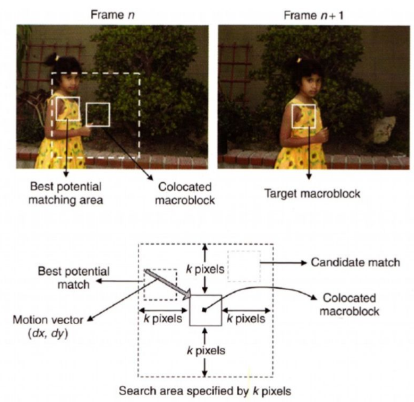

Figure 1. Motion Vector Computation

In this lab, a given target 16x16 sized block from the current frame will be tested against every possible candidate clock in a search region from the previous frame to determine the location of the best matching candidate block. For each candidate a metric is calculated to assess the 'similarity' between the candidate block and the target block. This metric is the Mean Absolute Difference (MAD) defined as follows:

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

where C_n is the pixel value from the candidate block in frame1, C_{n+1} is the pixel value from the target block in frame2 with location *(p,q), i* and *j* are the shifts applied to the location in frame1 when iterating through different candidate blocks, and *m* and *n,* are the dimensions of the target block.

B. Content Based Retrieval

Content-based image retrieval (CBIR) also referred to as query by image content is the application of retrieving images from a database that resemble closest to a query image using a set of descriptors. This experiment will utilize colour histograms as the descriptor. There will be 3 different metrics to determine the best matching colour histograms:

Manhattan (cityblock) distance:

$$(3)\ d_{L1}(h,g) = \sum_i |h(i) - g(i)|$$

Euclidean distance:

$$(4)\ d_{L2}(h,g) = \sum_i (h(i) - g(i))^2$$

Histogram Intersection:

$$(5)\ d_{int}(h,g) = \frac{\sum_i min(h(i), g(i))}{min(|h|,|g|)}$$

III. Methodology

A. Motion Vector Computation

In this lab, a random sample video was taken and loaded. The 10th and 20th frame was taken to perform motion vector computation and DPCM such that the two frames had some visible motion in it. The second frame was divided into 16x16 sized block and each one was compared against candidate blocks from its corresponding search area from frame 1. The function *computeMotionVec(framePrev, frameCurr, p, q, k)* was scripted to perform motion vector calculation where *(p,q)* is the coordinate of the top left pixel of the colocaled (target) block, and *k* is the search radius as visualized in Figure 1.

The *computeMotionVec()* function calculates the MAD values of the target block against every candidate block and records the distance in terms of x and y difference values (dx,dy). It then outputs the (dx,dy) of the candidate block with the least MAD value. These [dx,dy] values will be added on as a shift to the first frame as predictors when performing DPCM.

B. Content Based Retrieval

The histograms were computed in the HSV colour space assigning 30 bins for the H channel and 15 bins each for the S and V channels. Afterwards, the individual histograms were concatenated into one single histogram which was entered into a database with the correlating image name. The database is now initialized and ready to be used to find the best matching colour histograms using either 1 of 3 similarity metrics for an input image. The custom function to accomplish this is *CBR(img, database, equation)* where *img* is the query image, *database* is the selected database, and *equation* is a number from 3 to 5 correlating to the similarity metric equation to be used, previously mentioned in the theory section. The function will compare the input image's colour histogram with those in the database and displays all the images in the database from most similar to least where lower values of equations 3 and 4 represented more similar images, and lower values of equation 5 represented less similar images. If the input image is part of the database, it will skip evaluating itself.

Instead of matching images by the global histogram, the local histograms can be used as well. The images can be split into 8 equally sized blocks (4x2) then it is possible to repeat the process as the global histogram for each individual block. Allowing 30 bins for the H channel and 15 bins each for the S and V bins, the local histograms for each block is found then concatenated. The total number of bins for each image will then be 480 ([30+15=15] * 8 = 480). Another custom function *CBR_block(img, database, equation)* is used similar to the previous function with the only difference being is that this one accounts for the input image being divided into 8 equally sized blocks.

IV. Results

A.. Motion Vector Computation

Figure 2. frame1 (10th frame of the video file)

Figure 3. frame2 (20th frame of the video file)

Figure 4. Frame difference between frame1(with motion vector shift as predictors applied) and frame 2

Figure 5. Direct frame difference between frame 1 and frame2

TABLE 1. Entropy Values

Frame Difference		***Entropy***
frame1 (with motion vector shifted predictors)	frame2	**1.23**
frame1	frame2	**2.46**

B. Content Based Retrieval

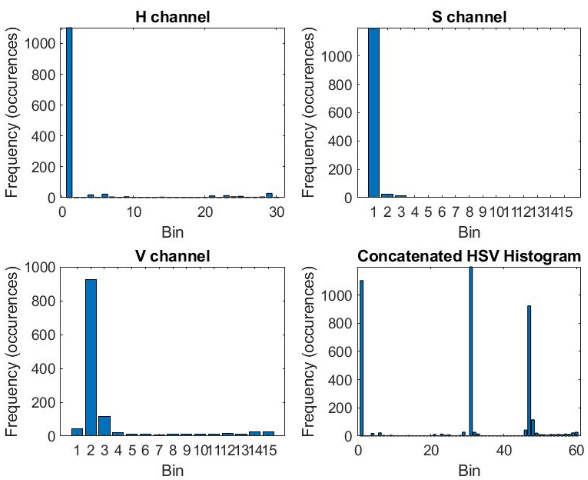

Figure 6.Separate and Concatenated HSV Histograms

Figure 7. Example Query Image

Figure 8. Example Results from Querying Database

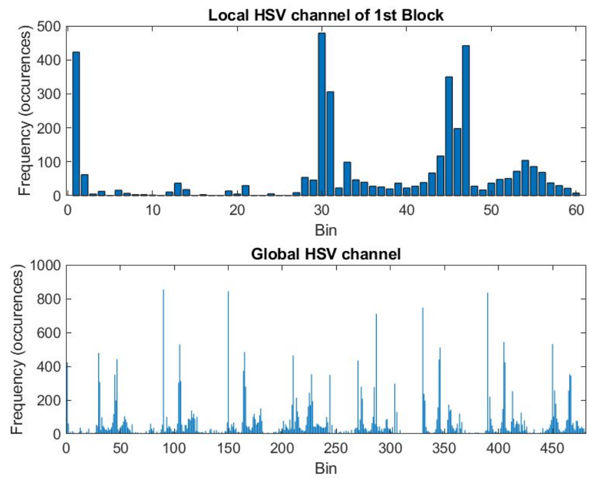

Figure 9. Local and Global Histograms

TABLE 2. Distance Metrics Using Global Histogram

	Distance Metric					
Image Number	**Manhattan Distance**		**Euclidean Distance**		**Histogram Intersection**	
	Value	**Order**	**Value**	**Order**	**Value**	**Order**
1	28242	3	60382596	5	15879	3
2	36912	8	67799600	7	11544	8
3	42488	11	82967094	9	8756	11
4	23952	1	67300844	6	18024	1
5	40608	10	92487902	11	9696	10
6	28094	2	43148364	1	15953	2
7	29246	5	52304224	2	15377	5
8	40244	9	88724444	10	9878	9
9	35782	7	69966454	8	12109	7
10	30058	6	53684468	3	14971	6
11	28416	4	59380148	4	15792	4

TABLE 3. Distance Metrics Using Local Histograms

Image Number	Distance Metric					
	Manhattan Distance		Euclidean Distance		Histogram Intersection	
	Value	Order	Value	Order	Value	Order
1	30820	2	9692964	2	14590	2
2	41198	9	11905102	5	9401	9
3	44680	10	14358430	9	7660	10
4	30586	1	12742590	6	14707	1
5	45650	11	16025236	11	7175	11
6	31214	3	8558084	1	14393	3
7	40010	6	14541476	10	9995	6
8	41184	8	13795674	8	9408	8
9	41072	7	12826168	7	9464	7
10	32574	4	10594808	3	13713	4
11	33170	5	11805174	4	13415	5

TABLE 4. Comparing the average order of images using

Image Number	Average Order	
	Global Histogram	Local Histogram
1	3.67	2
2	7.67	7.67
3	10.33	9.67
4	2.67	2.67
5	10.33	11
6	1.67	2.33
7	4	7.33
8	9.33	8
9	7.33	7
10	5	3.33
11	4	4.67

V. Discussion

A. Motion Vector Computation

Figure 2 and Figure 3 are the two frames analyzed in this lab. The motion is quite difficult to see but looking at Figure 4 and Figure 5, it is clearly seen where motion is most abundant. Figure 5 shows more white outlines compared to Figure 4 meaning that there is more pixel difference between frame1 and frame2 directly. However, when the motion vector shifts are applied to frame1 and their frame difference are calculated, there is visibly less white outlines meaning that there are not as much pixel value difference between them. This is because with motion vectors as predictors, we find more accurate representation where the 'same' group of pixels will be. With a decrease in frame difference, there will be a decrease in entropy. This expected results in seen in Table 1. This is ideal in most cases, since with less needed bits to represent an image, there is less memory required for processing and compression.

B. Content Based Retrieval

Figure 6 shows the individual histograms of the H, S, and V channels concatenated into one histograms with a total of 60 bins. Figure 7 shows the example query image used for image content based retrieval where Figure 8 shows an example of 3 images returned from the database in a order of most similar to least. Tables 2 show 11 test images with corresponding Manhattan, Euclidean and Histogram Intersection values along with the order using the global histogram descriptor. Images with lower order are more similar to the query image, with 1 being the best and 11 being the worst. For Manhattan and Euclidean, the lower the values were, the more similar the images were to the query image. For Histogram Intersection, the opposite was true where a higher value meant the images were more similar. Table 3 shows the results of the 3 distance metrics as well but using the local histogram method. In both methods, the Manhattan and Histogram Intersection gave the same order of the images but the Euclidean metric results differed. Table 4 which shows the average order of the test images using the local and global histogram methods.

VI. Conclusion

In conclusion, calculation motion vectors and applying these shifts as predictors when performing DPCM will result in smaller entropy values, meaning less bits and less memory required to represent inter-frame differences in a video. In the content based retrieval analysis, test images #1 and #4 were very similar to the query image, with all 3 images containing a red object in the center. From Table 4, tt can be seen that the order of images #1 and #4 are higher using the local histograms than the global histograms. No other images, were of similar nature with a red object in the middle, therefore, based on these results, it was more accurate to use the local histograms rather than the global histograms as descriptor for image based content retrieval.

References

[1] Z. Li and M. Andrew, *Fundamentals of Multimedia*, 1st ed. New Jersey: Pearson Education, 2004.