PROJECT 2 –Block Matching Algorithm and Motion Compensation

CS 443/543, Spring 2016   Instructor: Haeyong Chung

You may implement this project individually or as a group. The expectations from the project are the same regardless of the number of students implementing the project. Therefore, group projects are encouraged.

Project Description:
Basically consecutive frames in a video are similar. Pixels in two adjacent video frames are likely to have the same values in the same location. Exploiting temporal redundancy is one of the most essential techniques in video compression. The difference between the current frame and other frame(s) in the sequence can be coded, instead of coding every frame of the video independently as a new image. Block matching is the most effective approach to exploit temporal redundancy for video compression. The aim of this project is to implement and evaluate the block matching algorithms which enable to take advantage of temporal redundancy to compress video data.

The primary goals of this project are:

- To implement the block matching algorithms which enable motion estimation
- To use motion vectors in predicting and processing video frames
- To observe and understand how motion compensated processing differs from non-motion compensated processing.

In this project, we conduct the following steps of video compression based on Motion Compensation (MC):

1. Motion Estimation (Search motion vectors).
    a. Sequential Search
    b. 2D Logarithmic Search
2. MC-based Prediction (Use motion vectors in prediction and processing of images)
3. Derivation of the prediction error, i.e., the difference (Compare the difference between original frames and predicted/estimated frames).
4. Display all results including Motion Vectors and Differences

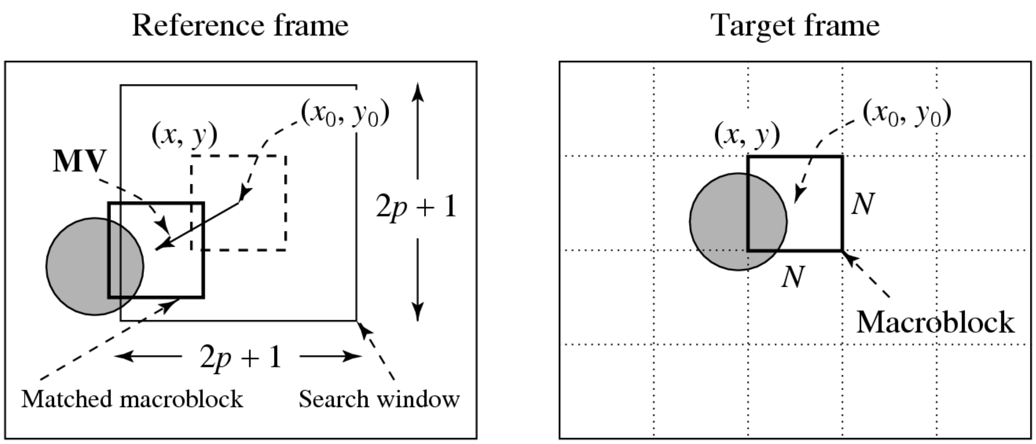
The block matching algorithms influence the overall quality of Motion Field and the computation time. This project has four parts.

Part I: Implement A Block Matching Algorithm (Check the ch.10 slides)

- Download and use **<u>two</u>** different frame samples, `twosampleframes.zip` from canvas (each sample has only two frames).
- The basic idea for block matching is as follows:
    - Each frame/image is divided into *macroblocks* of size N x N.
    - By default, $\boldsymbol{N = 16}$ for luminance images (We only use the luminance, i.e., gray channel of the image for this project). In this project we will use search parameter $\boldsymbol{p=7}$ *so* the search window size $\boldsymbol{M=15}$.
    - Motion compensation is performed at the macroblock level.
    - The current image frame is referred to as *Target Frame*.
    - A match is sought between the macroblock in the Target Frame and the most similar macroblock in previous and/or future frame(s) (referred to as *Reference*

frame(s)). For our sample files, use 'frame1' as the reference frame and use 'frame2' as the target (current) frame.

- The displacement of the reference macroblock to the target macroblock is called a *motion vector* **MV**.
- The following figure shows the case of *forward prediction* in which the Reference frame is taken to be a previous frame.



- The difference between two macroblocks can then be measured by their *Mean Absolute Difference (MAD)*:
-

$$MAD(i, j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \left| C(x+k, y+l) - R(x+i+k, y+j+l) \right|$$

N — size of the macroblock,
k and l — indices for pixels in the macroblock,
i and j — horizontal and vertical displacements,
$C(x+k, y+l)$ — pixels in macroblock in Target frame,
$R(x+i+k, y+j+l)$ — pixels in macroblock in Reference frame.

- The goal of the search is to find a vector (i, j) as the motion vector ***MV*** = (**u**, **v**), such that $MAD(i, j)$ is minimum:

$$(u, v) = \left[\ (i, j) \mid MAD(i, j) \text{ is minimum},\ i \in [-p, p],\ j \in [-p, p] \right]$$

- Based on this, you must implement a block matching/search approach: (1) Sequential Search or (2) 2D Logarithmic Search. The output of this part will be motion vectors. You can refer to the following pseudo codes for each algorithm.
    - **Sequential Search ('Pure' undergraduate team):** sequentially search the whole (2p + 1) * (2p + 1) window in the Reference frame (also referred to as full or brute force search).

```
BEGIN
     min_MAD = LARGE NUMBER;      /* Initialization */
     for i = -p to p
     {
       for j = -p to p
       {
              cur_MAD = MAD(i, j);
              if cur_MAD  <  min_MAD
              {
                     min_MAD = cur_MAD;
                     u = i; /* Get the coordinates for MV. */
                     v = j;
              }
       }
     }
END
```

- **2D Logarithmic search (If there is a grad student in your team, you must implement this, but undergrad teams can select this, too):**

```
BEGIN
     offset = ⌈p/2⌉;
     Specify nine macroblocks within the search window in the Reference
     frame, they are centered at (x0, y0) and separated by offset
     horizontally and/or vertically;
     WHILE last ≠ TRUE
     {
          Find one of the nine specified macroblocks that yields minimum
             MAD;
       IF offset = 1 THEN last = TRUE;
       offset = ⌈offset/2⌉ ;
       Form a search region with the new offset and new center found;
     }
END
```

Part II: Create a Motion-Compensated Frame (predicted/estimated image) using Motion Vectors Computed from Part I.

You may refer to an algorithm like this:

- ***Inputs:*** will be the reference image and motion vectors (and the MB size)
- ***Output:*** will be the motion compensated image (predicted/estimated frame)

1. Initialize the motion compensated (MC) image.
2. Start from the top left of the motion compensated (MC) image.
3. Divide the MC image into MacroBlock (MB) (N=16).
4. For each MB of the MC image, you need to read the corresponding motion vector generated from the previous part and;
5. Fill each MB of the MC image with the corresponding MB of the reference image (You can find it using the corresponding Motion Vectors).

When you're done with all the blocks in the MC image, then the MC image becomes the motion compensated version of the previous (reference frame).

Part III: Error/Difference Computation

The difference between the current frame and the prediction/MC frame from the previous step refers to Displaced Frame Difference (DFD) between each frame. You need to compute the two different differences:

1. Compute the Frame Difference (FD) between the target (current) frame I_n and reference (previous) frame I_{n-1}.

$$FD(i, j) = \left| I_n(i, j) - I_{n-1}(i, j) \right|$$

2. Compute the Displaced Frame Difference (DFD) between the target (current) frame and MC frame $\hat{I}_{n-1}$.

$$DFD(i, j) = \left| I_n(i, j) - \hat{I}_{n-1}(i, j) \right|$$

3. Compute the Peak Signal-to-Noise Ratio. You are also required to compute the frame distortion in terms of Mean Square Error and then calculate the PSNR value.

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[I_n(i, j) - \hat{I}_{n-1}(i, j) \right]^2$$

$$PSNR = 20 * \log_{10} \frac{255}{\sqrt{MSE}}$$

Part IV: Display

1. Display the target frame, reference frame, and MC frame.
2. Display (visualize) Motion Vectors for the corresponding MBs and Superimpose them on the target image.
3. Display FD and DFD images (You may need to remap FD/DFD values to a specific range in order to display them appropriately).
4. Display the MSE and PSNR.

Other Conditions:

- Use only the luminance channel for this project. If an image is color, first change it to a grayscale image.
- Download and use only provided image sample. For our sample files, use 'frame1' as the reference frame and use 'frame2' as the target (current) frame.
- You must run from Part I to IV for two different macroblock sizes:
  1. ***N* = 16** (i.e., 16by16 macroblock size); Search parameter ***p=7;   M=15***.
  2. ***N* = 8 (**i.e., 8by8 macroblock size); Search parameter ***p=7;   M=15***.

Submitting Materials

You should submit the following materials for the grading.

- Source code
- General description of your implementation and user interface (at most 3 pages).
- Your (group) report.
  - Your report should include all results and images required from Part IV.
  - Your report should include some discussion about a comparison of FD and DFD results.
  - Also, your report should include some discussion about a comparison of two settings (N=16 vs. N=8) in terms of MSE and PSNR. Which macroblock size provides better quality? What is the tradeoff between two sizes?
  - On your report, you must include all team member's name and each team member must submit it individually.

Important Dates:

Report and Source code: April 20th (Canvas)

5-min Presentation: April 21th (in class)