ECE462 Multimedia Systems

Laboratory Assignment 4: Video Encoding
Spring 2019

TA: Keerthi Nelaturu

Procedure - Week 1

All work from this procedure must be demonstrated to the TA by the end of the the first Lab 4 session.

1 Motion Search Algorithm Implementation

The lab preparation handout provided you with some basic information about video encoding. In this lab, you are to implement two motion search algorithms, namely Sequential Search and Logarithmic Search. **You must familiarize yourself with these search algorithms by reading the textbook sections 10.3.1 and 10.3.2.**

It is important that you verify that your motion search algorithms work correctly because they will be used in the next week of the lab.

Requirements for each motion search implementation:

- Implement the function as follows:
  `[Prediction, MotionVectors, Flag]=<type>_MotionSearch(Target, Reference)`, where:
  - `<type>` is either `Sequential` or `Log` (you must implement both).
  - The inputs `Target` and `Reference` are $M \times N$ matrices representing the current target picture and reference picture, respectively.
  - The output `Prediction` is the $M \times N$ *predicted picture*.
  - The output `MotionVectors` is a $K \times 2$ matrix for *motion vectors*, where K is the number of MBs in the picture – the MVs should be ordered based on scanning the MBs of the current frame from left to right, top to bottom. See Fig. 1 for an example where $K = 12$.
  - The output `Flag` is $K \times 1$ matrix to *indicate* whether the motion search is successful for the corresponding MB (i.e., if the candidate reference MB with the lowest cost is below the threshold – see below regarding the threshold). Use a “1” to indicate that the motion search was successful; a “0” for an unsuccessful search.

1	2	3	4
5	6	7	8
9	10	11	12

Figure 1: MB scanning order. $K = 12$ example – in general, you will have a different number of MBs.

- You will use an MB size of 16×16 and you will only be dealing with one (luminance) channel.

- Use sum of absolute difference (SAD) as the cost measure:

$$\mathrm{SAD}(i, j) = \sum_{k=0}^{15} \sum_{l=0}^{15} |T(x + k, y + l) - R(x + i + k, y + j + l)|$$

  where T is the current (target) frame, R is the reference frame, (x, y) is the upper-left corner of the current MB in the target frame, and (i, j) is the candidate motion vector (the relative coordinates, or offset of the reference MB being tested).

- Use an SAD cost threshold of 2048. If all the candidate MBs tested in the reference have an SAD greater than the threshold, the motion search is considered to be *unsuccessful*. Remember to indicate whether the search was successful in the $K \times 1$ vector output described above. For an unsuccessful search, set the corresponding MV to $(0, 0)$, and predicted values to all zeroes (0) for the MB.

- If there is a tie in SAD between two motion vectors, you can use either as the final MV (in practice, you would generally use the one producing the smallest MV).

- Set the search area to be ± 8 pixels in each direction (i.e., $p = 8$).

Hints:

- Implement your SAD as a separate function so that your motion search can just pass off the current target MB and the reference MB being tested to get the cost measure.

- Remember the correct MB scan order (Fig. 1). You would generally have two nested loops to iterate through the MBs in the target frame. To produce the correct order, your outer loop would iterate through the rows (first matrix index in MATLAB) and the inner loop would iterate through the columns (second matrix index in MATLAB).

- For each of the two different search algorithms (sequential and logarithmic) it may be useful to write a function that returns a list of locations to search, according to the current target MB coordinates. For example, for a target MB with (x, y) (top-left corner) coordinates $(17, 17)$ (using MATLAB coordinates), the sequential search reference block locations (again, top-left corner) are $\{(9, 9), (9, 10), \ldots, (9, 25), (10, 9), \ldots, (10, 25), \ldots, (25, 25)\}$. For the logarithmic search, your function may take as input the offset of the current iteration (rather than p) – given the first iteration has an offset of $\lceil p/2 \rceil$, the first iteration reference locations for a MB at $(17, 17)$ would be $\{(17, 17), (13, 17), (21, 17), (13, 13), (17, 13), (21, 13), (13, 21), (17, 21), (21, 21)\}$.
- Remember to exclude reference locations outside the reference frame boundaries. In MATLAB coordinates, this means that the reference coordinates cannot be less than $(1, 1)$ and cannot be greater than $(M - 15, N - 15)$ (remember, the coordinates are for the top-left corner of the block).
- Remember that an MV is the *offset* or difference between the target MB coordinates and the best match reference coordinates. For example, if our target MB is at $(17, 17)$, and our best match in the reference frame is at $(13, 21)$, then the output MV is $(-4, 4)$.

2 Procedure

Load the images for your lab section: `lab4_wk1_<section>_tgt.pgm` is the current target picture; `lab4_wk1_<section>_ref.pgm` is the reference picture.

1. Run your sequential search algorithm:
   (a) $\boxed{\text{4 Marks}}$ Display the motion vectors using the provided function, `show_mv(MotionVectors)`, where `MotionVectors` is the $K \times 2$ matrix of MVs output from your search function.
   (b) $\boxed{\text{5 Marks}}$ Display the predicted picture and the residual picture (difference between target picture and predicted picture). **Note: since the residual picture can have positive and negative values, you should convert your target and predicted picture to type `double` before taking the difference, and display it using `imagesc` and an appropriate gray map.**
   (c) $\boxed{\text{1 Marks}}$ Calculate the MSE of the predicted picture.
2. Run your logarithmic search algorithm:
   (a) $\boxed{\text{4 Marks}}$ Display the motion vectors using the provided function, `show_mv(MotionVectors)`, where `MotionVectors` is the $K \times 2$ matrix of MVs output from your search function.

(b) $\boxed{\text{5 Marks}}$ Display the predicted picture and the residual picture (difference between target picture and predicted picture). **Note: since the residual picture can have positive and negative values, you should convert your target and predicted picture to type `double` before taking the difference, and display it using `imagesc` and an appropriate gray map.**

(c) $\boxed{\text{1 Marks}}$ Calculate the MSE of the predicted picture and compare with the value obtained using the sequential search algorithm.