

CSE 4084 Multimedia Systems
Homework 2

Problem 1

In this problem you will perform block matching motion estimation between two consecutive video frames. Follow the instructions below to complete this problem.

- (1) Download the two video frames from files `frame_1` and `frame_2` files. The frames/images are of height 288 and width 352.
- (2) Load the frame with the file name “frame1.jpg” into a 288 x 352 MATLAB array using function “`imread`”, and then convert the array type from 8-bit integer to real number using function “`double`” or “`cast`” (note that the range of intensity values after conversion is between 0 and 255.) Denote by I_1 the converted MATLAB array. Repeat this step for the frame with the file name “frame2.jpg” and denote the resulting MATLAB array by I_2 . In this problem, I_2 corresponds to the current frame, and I_1 corresponds to the previous frame (i.e. the reference frame).
- (3) Consider the 32 x 32 target block in I_2 that has its upper-left corner at (65,81) and lower-right corner at (96,112). Note this is MATLAB coordinate convention, i.e., the first number between the parenthesis is the row index extending from 1 to 288 and the second number is the column index extending from 1 to 352. The target block is therefore a 32 x 32 sub-array of I_2 .
- (4) Denote the target block by B_{target} . Motion estimation via block matching searches for the 32 x 32 sub-array of I_1 that is “most similar” to B_{target} . Recall that in the lectures we have introduced various forms of matching criteria e.g. correlation coefficient, mean-squared error (MSE), mean-absolute-error (MAE), etc.

In this problem, we use MAE as the matching criterion. Given two blocks B_1 and B_2 both of size $M \times N$, the MAE is defined as

$$MAE(B_1, B_2) = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N |B_1(i, j) - B_2(i, j)|$$

To find the block in I_1 that is most similar to B_{target} in the MAE sense, you will need to scan through all the 32 x 32 blocks in I_1 , compute the MAE between each of these blocks and B_{target} , and find the one that yields the smallest of MAE.

Note that in practice motion search is only performed over a region of the reference frame, but for the sake of simplicity, we perform motion search over the entire reference frame I_1 , in this part. Determine the coordinates of the upper-left corner of the matched block in MATLAB convention. What is the backward motion vector for the center pixel of B_{target} ?

- (5) What is the corresponding MAE value (up to two decimal points)?
- (6) Find and show the frame difference ($DFD = I_2 - I_1$). Estimate the backward motion vector of each block in frame I_2 by using 16 x 16 blocks and searching for the best

matching block in I_1 . Use a search region of size 64x64. Plot the motion vectors on frame I_2 using the MATLAB function **quiver()**.

Predict frame 2, i.e., \hat{I}_2 by motion compensating the blocks in frame I_1 . Find and show the displaced frame difference (DFD = $I_2 - \hat{I}_2$) (similar to the images in slide 57 of motion estimation lecture notes). Compute the PSNR value between the original image I_2 (converted to type 'double') and the predicted image \hat{I}_2 .

- (7) Repeat step (6) using logarithmic search and compare your results. What are the advantages/disadvantages of using logarithmic search?
- (8) Discuss the advantages and disadvantages of the block-matching based motion estimation method.

Problem 2

In this problem you will perform median filtering to enhance the quality of a noise corrupted image. Recall that median filtering is effective for removing "salt-and-pepper" noise from images. Follow the instructions below to complete this problem.

- (1) Download the noisy image. Load the noisy image into a MATLAB array and convert the type of the array from 8-bit integer 'uint8' to real number 'double'. Visualize the noisy image using the built-in MATLAB function "imshow". The function "imshow" takes as its argument either [0,255] for an 8-bit integer array (i.e. of type 'uint8'), or [0-1] for a normalized real-valued array (i.e. of type 'double'). To provide imshow with the correct argument, you would need either "cast" your real-valued array into "uint8", or normalize it by 255.
- (2) Perform 3x3 median filtering using the built-in MATLAB function "medfilt2". For this problem, the only argument you need to provide "medfilt2" with is the array you have created in step (1). Visualize the filtered image using "imshow". Remember to either cast the result to 'uint8' or normalize it before feeding it to "imshow".
- (3) Perform a second-pass median filtering on the filtered image that you have obtained from step (2). Visualize the two-pass filtered image. Compare it with the noisy input image and the 1-pass filtered image.
- (4) Download the noise-free image. Compute the PSNR value between the noise-free image and the noisy input image (up to two decimal points).
- (5) Calculate the PSNR value between the noise-free image and the 1-pass filtering output (up to two decimal points).
- (6) Calculate the PSNR value between the noise-free image and the 2-pass filtering output (up to two decimal points).

Submission Instructions:

1. Each student should submit his/her own homework. You can discuss the questions with your friends, but you must write your own solutions and code. Group work is not allowed. You can not exchange any written material, code or pseudocode. This also includes material found on the web.
2. Write a detailed report using Word/Latex and convert it to a pdf file. The report must include explanations about each part in each question. If you solve some of the questions by pencil and paper, you can scan your solution and insert it to your report (you can use free Mobile applications such as Adobe Scan).
3. Explain how your scripts and functions work, i.e., which parts of your functions/scripts accomplish which task and how it is accomplished. Include the outputs of your functions and figures to your report. Each figure should have a caption and should be explained in the text.
4. You can use MATLAB or Python for programming assignments. Don't forget to put detailed comments into your functions/scripts to explain what your code is doing. Also indicate the inputs and outputs in the comment section.
5. Combine your report and codes into a single zip file. Plots and figures should go into the report.
6. Name your zip file as "CSE4084_name_surname_hw_no.zip". For example, a student whose name is Ayşe Çalışkan will name her file as: "CSE4084_ayse_caliskan_hw1.zip" for the first homework. Also, write your name, surname and student number as comments at the beginning of your codes.
7. Submit your homework via the class web page at classroom.google.com before the deadline.
8. Late submissions will lose 10 points for each day after the deadline.