

Programming on Block Based Motion Compensation (80 points)

This programming assignment will help you gain an understanding of issues that relate to block-based motion compensation. *Note – use of any external libraries is not allowed, you are expected to explicitly implement the details as instructed below.*

Given two consecutive image frames – frame_n and frame_{n+1} , write a function that creates and display two images as follows

- a *predicted or reconstructed frame* for frame_{n+1} built using motion compensation techniques from frame_n and,
- error difference frame.

You will be using the full brute force method, by dividing frame_{n+1} into 16×16 blocks and performing motion compensation within an input search area k . The details of the algorithm have been explained in the class lecture and in the textbook. Here is how your program will be invoked

MyMotionPredictor.exe frame_n.rgb frame_{n+1}.rgb k

Input to your program will be three parameters - two consecutive rgb image frames of size 640×320 (**first parameter is previous frame, second parameter is frame to predict**) and search parameter k , which will define the search area to search into, k can have values from 1 to 32. Speed is important, but more important is the accuracy of your output.

Although your inputs are .rgb frames, you will need to compute motion vectors only on the Y channel – so your process will produce and display two gray level images.

Conversion of RGB to YUV

Given R, G and B values the conversion from RGB to YUV is given by

$$\begin{array}{rcl} Y & = & 0.299 \quad 0.587 \quad 0.114 \quad R \\ U & & 0.596 \quad -0.274 \quad -0.322 \quad G \\ V & & 0.211 \quad -0.523 \quad 0.312 \quad B \end{array}$$

Remember that if RGB channels are represented by n bits each, then the YUV channels are also represented by the same number of bits. In this assignment you are asked to use just the Y value for doing motion prediction.

Example output is shown below



predicted Y channel for frame_{n+1}



error difference for Y channel