EECS2301E

Lab 6

Fall 2018

Lab Objectives

In this lab you will learn how to calculate the minimal shift of a vector to minimize the difference with another vector. Then we will generalize this to the motion estimation problem in video compression.

Problem 1

Write a C program that calculates the best rotation for vector $\vec{Y}$ to minimize the difference with $\vec{X}$.

Given two vectors $\vec{X}$ and $\vec{Y}$ of length *N*, where *N* is limited to 100.

The difference between these two vectors is (| | means the absolute value)

$$\Delta=\sum_{i=0}^{N}|\ x[i]-y[i]\ |$$

We would like to rotate y by a specific number in order to minimize the difference. The rotation is circular rotation. What is that number? In case of tie, we choose the shortest shift/rotation.

The input is as follows (from the standard input

the vector length (N)

vector x

vector y

For example

5

1 4 7 23 17

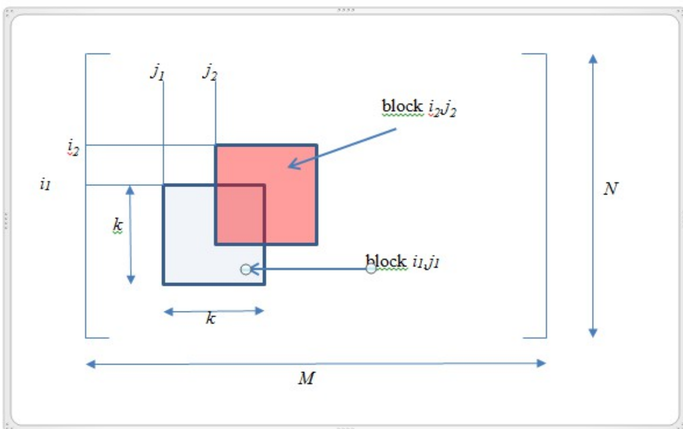
23 4 5 6 77

The output is a single number between 0 and 4 (since the length is 5, a shift of 5 is the same as a shift of 0)

submit 2031 lab_6 a1.c

Problem 2

This is a generalization of the above question for 2-D arrays. In video compression this is known as motion estimation. The video is divided into frames, each frame is divided into blocks (for example the frame could by 1000$^{3}_{2}$1000, and the block is 10$^{3}_{2}$10). That means every frame is 10,000 blocks. If I know that block number x in the current frame moved to location y in the next frame, so instead of sending the 10$^{3}_{2}$10 block, I will send "moved to Y" which is a lot less data than the entire block. The point is to calculate Y (or Y-X to be exact).



Consider the above picture that shows a frame M$^{3}_{2}$N and two sub-blocks, one sub-block with its upper left corner at i_1,j_1 and the second one at i_2,j_2. The second block is a translation of the first block by i_2-i_1 in the i direction, and j_2-j_2 in the j direction. The difference between these blocks is (there are many ways to calculate this, we will use the easiest one which is the sum of absolute errors.

$$\Delta=\sum_{i=0}^{k-1}\sum_{j=0}^{k-1}|\ x[i+i_1][j+j_1]-x[i+i_2][j+j_2]\ |$$

Now how does that work.
Starting with the first block i_1,j_1 Look for the block with minimum distance by moving this block in the vertical as well as the horizontal direction. First we move the block in the i direction from $i_1-\delta$ to $i_1+\delta$ and in the j direction from $j_1-\delta$ to $j_1+\delta$, where δ ranges from $1\rightarrow z$. If z is 3, then we consider -3, -2, -1, 1,2,3 for both vertical and horizontal direction for a total of 36 different block. We calculate the difference between the original block and these 36 blocks and we choose the one with the minimum distance.

In this lab, you are given the frame size (M,N), the block size (k), the block position (i_1,j_1) and the maximum distance for search (z). Calculate the block with the minimum difference with the specified distance.

Input file
M N
k
z
i_1 j_1
matrix size M-by-N row by row

Output
i2-i1 j2-j1
difference.

Example
8 8
3
2
2 1
1 2 3 4 5 6 7 8 9 10 11 12 3 4 5 6 7 8 … (64 elements)

submit 2031 lab_6 a2.c