

Parallel implementation of Submatrix Search

Problem Definition

Picture(N) and **Object(N)** – are square matrices of integers with N rows and N columns. The project deals with a sets of Pictures and Objects of different sizes. Each member of the matrix represents a “color”. The range of possible colors is [1, 100].

Position(I, J) defines a coordinates of the upper left corner of the Object into Picture.

For each pair of overlapping members **p** and **o** of the Picture and Object we will calculate a relative difference

$$\text{diff} = \text{abs}((\mathbf{p} - \mathbf{o})/\mathbf{p})$$

The total difference is defined as an average of all relative differences for all overlapping members for given Position(I, J) of the Object into Picture. We will call it **Matching(I, J)**.

For example, for the Picture and Object from the Fig.1 the matching at Position(0,0) is equal

$$\text{Matching}(0,0) = (\text{abs}((10-5)/10) + \text{abs}((5-14)/5) + \text{abs}((67-9)/67) + \text{abs}((23-20)/23) + \text{abs}((6-56)/6) + \text{abs}((5-2)/5) + \text{abs}((12-6)/12) + \text{abs}((10-10)/10) + \text{abs}((20-3)/20))/9$$

10	5	67	12	8	4
23	6	5	14	9	5
12	10	20	56	2	3
1	2	6	10	3	2
45	3	7	5	5	2
11	43	2	54	1	12

5	14	9
20	56	2
6	10	3

Fig 1. Picture (6) and Object (3)

10	5	67	12	8	4
23	6	5	14	9	5
12	10	20	56	2	3
1	2	6	10	3	2
45	3	7	5	5	2
11	43	2	54	1	12

Fig 2. Ideal Matching of the Object into the Picture at Position (1, 2)

This project finds if the given picture contains at least three objects from the given object set.

Input data and Output Result of the project

The input file contains in the first line a Matching Value. Next line contains a number of Pictures in the file. For all pictures the first line defines its ID, next line contains the picture size, followed by picture elements, row by row. The same order is for objects as well.

Input.txt

Matching value

Number of Pictures

#For each picture:

Picture ID

Picture dimension(N)

N lines of members of the picture, row by row

...

Number of Objects

#For each object:

Object ID

Object dimension(N)

N lines of members of the object, row by row

Output.txt

The output file contains information about result for each Picture - if there were found at least three Object with a Matching value less than given one. It has to be in the following format:

In case some object was found

Picture Id: found Objects: ID1 Position1 (I,J) ; ID2 Position2 (I,J) ; ID3 Position3 (I,J)

In case NO three objects were found into the Picture

Picture Id: No three different Objects were found

For example,

Picture 1: found Objects: 3 Position(1,2) ; 6 Position(11,24) ; 3 Position(6,7)

Picture 2: found Objects: 5 Position(1,2) ; 8 Position(11,24) ; 3 Position(6,7)

Picture 3: No three different Objects were found

Requirements

- Implement the explained above Simplified “Recognition” algorithm to find an Object into Picture with appropriate Matching value.
- Input file contains a number of Pictures and Objects. For each Picture find if there exist at least three objects with an Appropriate Matching. The search for the given Picture is stopped when a different three Object from the object set are found.
- The input file **input.txt** initially is known for one process only. The results must be written to the file **output.txt** by the same process.
- The computation time of the parallel program must be faster than sequential solution.
- Be ready to demonstrate your solution running on VLAB (two computers from different pools when using MPI)