By Stela Carneiro Espíndola

*Short Explanation:*

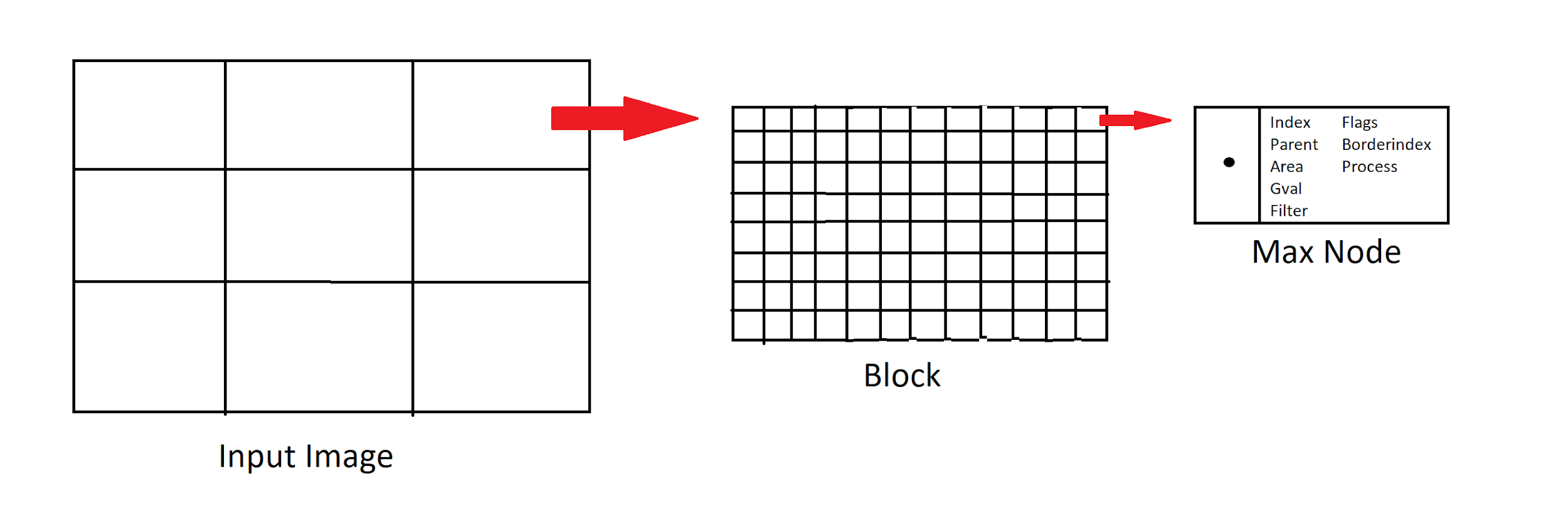
1. MaxTree
2. MaxNode
3. Boundary-tree
4. Merging tiles (block)
5. Obtaining attributes of the full tree

MaxTree:

To process large images that could not be entirely loaded into the memory, they divided the input image into blocks. Each block was turned into a MaxTree, and in each node of the MaxTree (MaxNode - one pixel) they put some informations about the values of that pixel.

The MaxTree is built according to the work of Wilkinson (*Concurrent computation of attribute filters on shared memory parallel machines*)

Max Node:

index → its index in the array

parent → the index of its parent

area → the area of all child nodes pointing to this node. ← This is the attribute filtered

gval → gray scale value of the pixel

filter → gray value after filtering

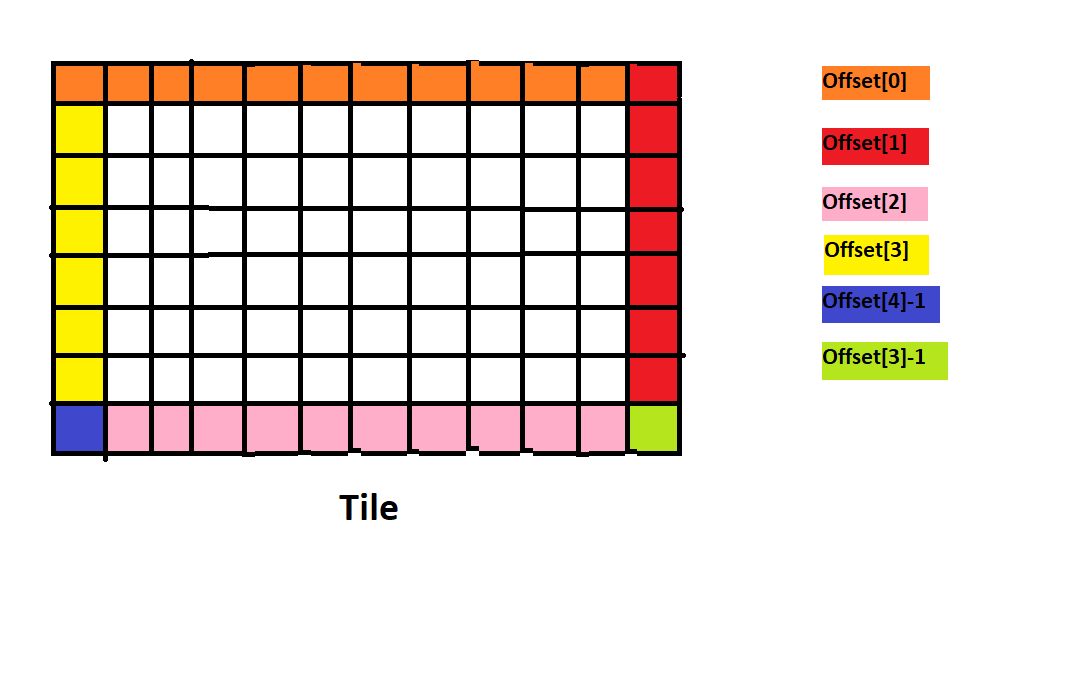
flags → reached, added (in the boundary-tree), changed

borderindex → the index in the array of the boundary-tree

process → the rank of the processor where the max-tree was built

Boudary-tree:

For each MaxTree, they compute a boundary-tree which is going to be send to the other processors to be merged. The boundary-tree is an 1D array with all of the nodes from the border and their parents. The data is produced as below:



array → all the MaxNodes in the border

offset → array of five offsets (north - 0, east - 1, south - 2, west - 3) and ancestors respectively

size → the number of elements in the boundary

border parent → an array with the index of this node in the boundary-tree and a pointer to which boundary-tree of the parent of this node belongs

merged → an array of booleans of size P indicating which processors are merged into this boundary tree

The boundary-trees are merged with each other:

Merging Tiles:

To merge two tiles we take the two boundary-trees and merge the nodes in a similar way as the sequential memory algorithm.

Basically:

1. The parent of the current node to merge is null; in this case we are done with the transversal for this branch; continue to the next node in the border;
2. The parent of the current node to merge is not in the merged tree yet; add this node by accumulating the area of this node, and point to its parent; then continue with the parent
3. The parent of the current node to merge is already in the merged tree. Add the accumulated area of this node to that node, and point to it; Then traverse its parent to add the accumulated area.

The important thing is: before merging the two trees, the processor sends the information about the changed nodes back to their original processors; then every processor will be updated;

After the merge of two boundary-trees a new one is created and the is sent to the next processor;