Algorithm 1 describes ‘Basic-NPS’ policy. The first step consists on asking operating system to determine the number of available processor. Depending on this number, algorithm will generate process. One active process will be assigned for each available processor. These new processes will belong to the SHED_FIFO class in order to ensure preemption and especially to avoid context switching. Process will only stop running if work is complete or less frequently when another process, belonging to the same class, with higher priority requesting processor. The global execution will stop if there no more task to process.

3.2.3 Merging phase

The key problem of each parallelization is merging obtained results. Normally this phase is done at the end of the process when all results are returned by all threads what usually means that only one output variable is declared and shared between all fighting threads. But as we mentioned in section 3.1, we are dealing with a dynamic evolution and if we take into account different steps of simple point detection then pixel characterizations, we can plan the following: The original shared data structure, containing all pixels, is divided into $n$ research zones $\{z_1, z_2, ..., z_n\}$. We associate one thread from the following list $\{T_1, T_2, ..., T_n\}$ to each zone. Each thread can browse freely its zone and if it detects target pixel types, it lowers characterized pixel and it pushes its eight neighbors in one of the available FIFO queues. A queue is said available if only one thread (owner) is using it. One queue cannot be shared by more than two threads so if no queue is available, threads can create a new one and become owners.

Since two threads finished, they directly merge and a new thread is created and then same process is lunched.

Fig.4: Illustration of merging phase with four threads