|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  37th Meeting: Geneva, CH, 4–10 October 2019 | Document: JCTVC-AK0024 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Advanced wavefront-based parallel solution decoding and encoding**  **for MV-HEVC** | | |
| *Status:* | Input document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Yong Beom Cho, Wei Liu Address Department of Electronic Engineering, Konkuk University, Seoul, South Korea | Tel: Email: | +82-10-9888-1423 ybcho@konkuk.ac.kr |
| *Source:* | Konkuk University | | |

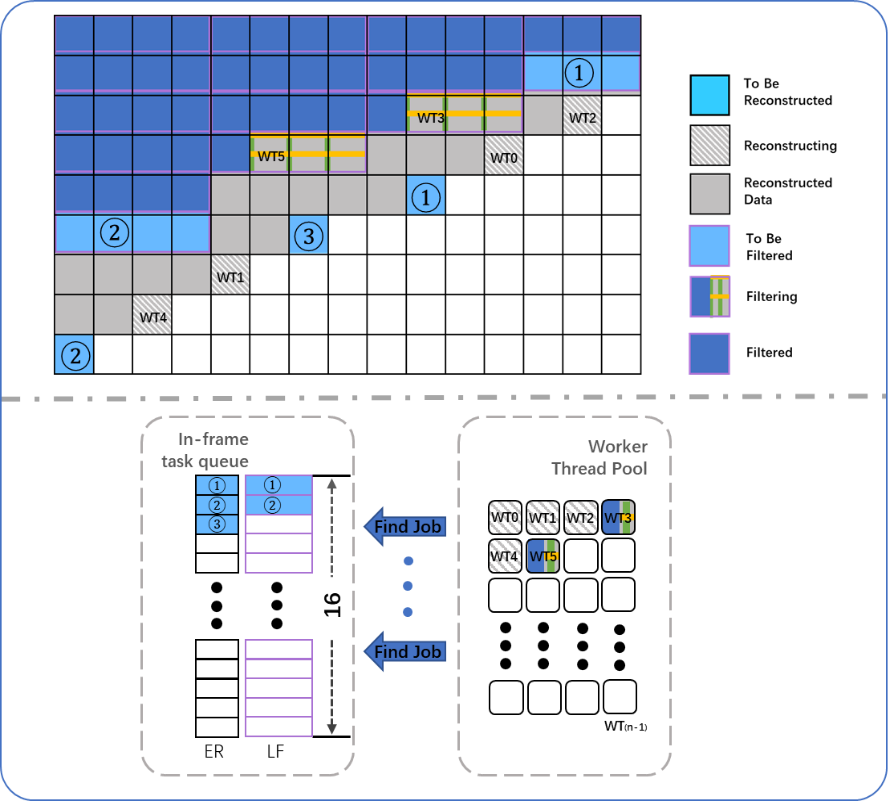
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

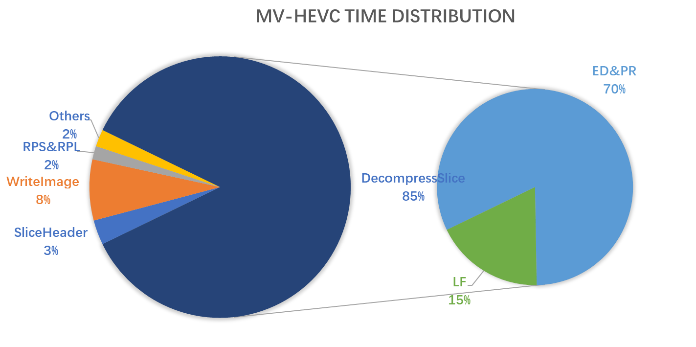
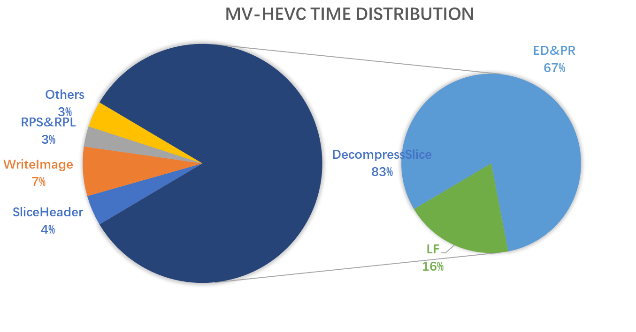
This paper presents an advanced Wavefront Parallel Processing method which is proposed to achieve higher intra-frame parallelism.

# Introduction

Our proposed approach is to complete tasks in parallel. In each working thread (WT), only the existing task unit needs to be completed to implement the same parallel mode of the decoded row, similar to WPP. The entropy decoder reconstruction pixel (EDRP) task unit is a CTU, and the loop filter (LF) task unit is several CTUs. In order to improve the in-frame parallelism, the EDRP and LF tasks are performed in a pipeline manner, and the LF is processed similar to the WPP method in parallel. In order to satisfy the data-dependent LF task, the previous row is delayed by one LF unit. The LF task is one LF unit later than the entropy reconstruction (ER) task to balance the complexity of processing the EDRP and LF task units without additional computation and schedule tasks between threads. This method reduces the computational load on the CPU. Another advantage is that, when an ER task has higher complexity because its own CTU block needs to be decoded, the following CTU row is waiting. Hence, the WT can perform LF processing. Because of the similar complexity, the complex CTU block decoding is completed and returned to the thread pool while the LF work is being completed. During this time, the CTU waiting for the ER task can continue with using the WT. This method also effectively improves the parallelism within the frame. Fig. 2 shows the time consumption of each function module of MV-HEVC. We found that the EDRP and LF occupied more than 80% of the time. Our analysis showed that the computational complexities of EDRP and LF are related to a certain relationship. The coefficient β is approximately equal to 4. Fig. 1 shows the LF task unit was set to four CTU blocks. When there are less than four CTUs at the end of each line, it is also an LF task unit. The size of the LF task unit can be freely set by adjusting the value of β.



**Fig. 1** ER, LF task assignment and execution sequence.



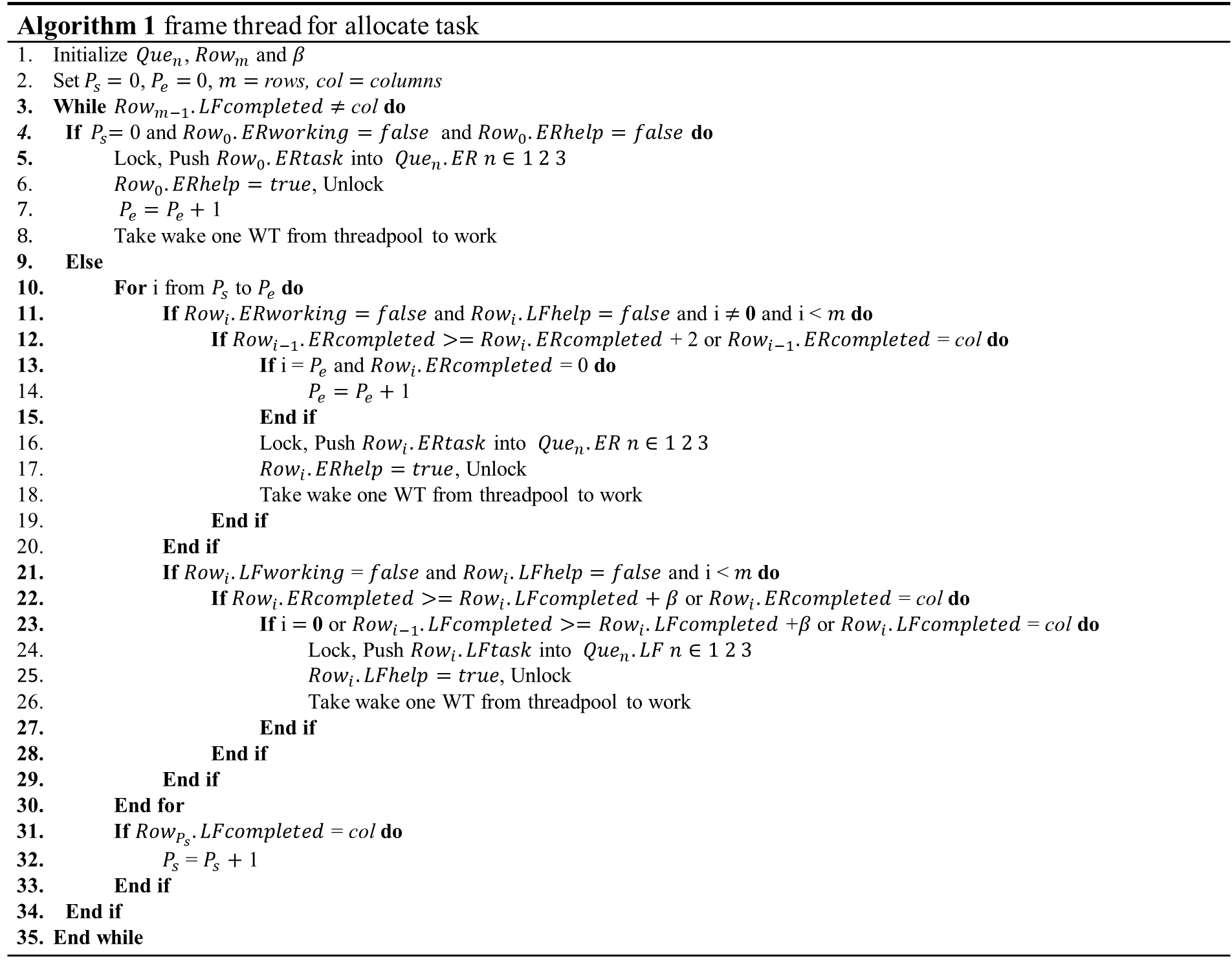
**Fig. 2** Profiling results for MV-HEVC.

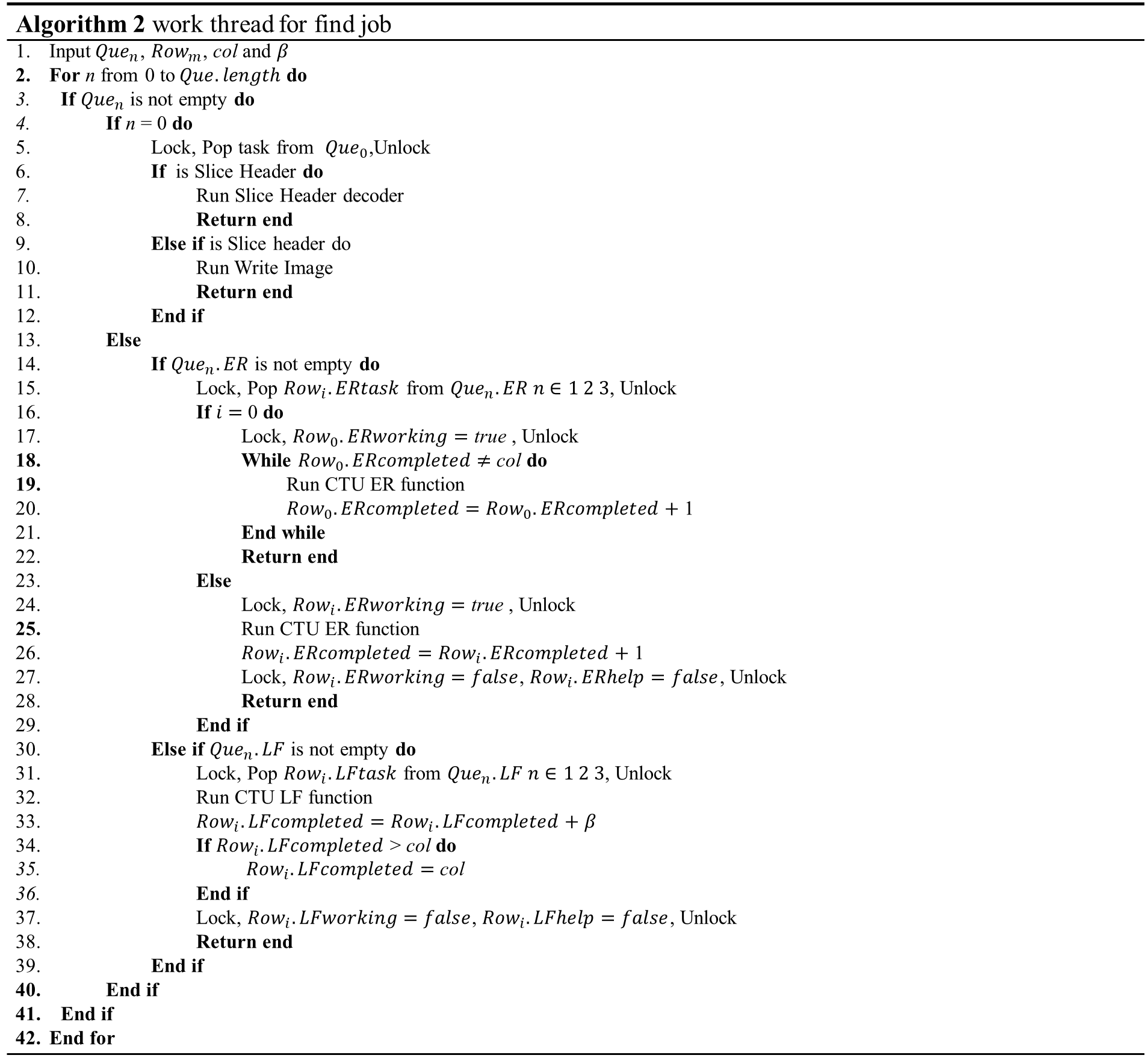
# Proposed parallel decoding

The proposed parallel method for intra-frame implementation is given in Algorithms 1 and 2. Algorithm 1 generates the MT task queue and wakes up the WT. Algorithm 2 searches the WT and performs the computation work. The FT is responsible for initializing the work, adding the first row of ER tasks to the ER queue, and waking the WT to work. The FT also monitor the computing units that can be queued in the parallel region of each frame. The WT first searches for tasks in the ER queue. If it is the first row, it decompresses the current line. If it is not the first row, after it executes the CTU task of the current row, it returns to the thread pool and waits to be awakened for the task. If the ER queue is empty, then the thread is woken up in the LF queue. This ensures that the ER task has a higher priority than the LF task so that the former is always executed first. This ensures that enough reconstructed CTUs are executed by the LF.

**Table 1** Variables for the pseudo code.

|  |  |
| --- | --- |
|  | Task queue 𝑛 ∈0 1 2 3 (v\_(n ) = 3) |
|  | Entropy decoder & Reconstruction pixel task queue |
|  | Loop Filter task queue |
|  | CTU rows structure |
|  | Completed ER task of CTUs in current row |
|  | Completed LF task of CTUs in current row |
|  | ER working state of current row |
|  | LF working state of current row |
|  | State of ER task pushed into task queue in current row |
|  | State of LF task pushed into task queue in current row |
|  | The number of CTUs with LF unit |
|  | The number of rows with a frame |
|  | The number of CTUs with a Row |
|  | Start row of parallel area |
|  | End row of parallel area |





# AWPS for encoder

The MT includes encoding the slice header, reference picture set (RPS), and RPL while writing images to play video or inter-frame YUV writing. There is no data dependency for slice header encoding between multiple frames. Therefore, multiple slice headers can be encoded simultaneously, and multiple slice header tasks can be performed at the same time. In order to avoid thread locks between the reference frame management task and YUV input/output (I/O) task, only one task is allowed in the task queue. The MT is also responsible for waking up the FT, and the FT thread sends the computation tasks to the in-frame queue. The MT is responsible for not only generating the task queue but also waking up the WT in the WTP. The WT searches for the task unit that needs to be processed from the task pool. The order in which the WT searches for the task is from Q0 to Q3 to ensure that the tasks in the MTQ are executed first. Moreover, the tasks in Q1 of the FTQ are always completed first. After Q1 is completed, Q2 replaces Q1, and Q3 replaces Q2. The priority of Q1 becomes Q3, and the FTQ cycle changes the priority order. Only one picture is reconstructed at the same time, so there is only one RPS, RPL, and image writing (IW) task in the MTQ. After the WT executes its own task, it returns to the thread pool and waits to be woken up to execute the next task. As long as the number of existing tasks is greater than the number of WTs, all of the computing resources of the CPU are utilized, which improves the parallelism and thus the decompression speed. Our proposed approach is to complete tasks in parallel. In each WT, only the existing task unit needs to be completed to implement the same parallel mode of the encoded row, similar to WPP. The Inter/Intra prediction and entropy encoder (PE) task unit is a CTU, and the loop filter (LF) task unit is several CTUs.

In order to improve the in-frame parallelism, the PE and LF tasks are performed in a pipeline manner, and the LF is processed similar to the WPP method in parallel. In order to satisfy the data-dependent LF task, the previous row is delayed by one LF unit. The LF task is one LF unit later than the PE task, to balance the complexity of processing the PE and LF task units without additional computation and schedule tasks between threads. This method reduces the computational load on the CPU. Another advantage is that, when an PE task has higher complexity because its own CTU block needs to be encoded, the following CTU row is waiting. Hence, the WT can perform LF processing. Because of the similar complexity, the complex CTU block encoding is completed and returned to the thread pool while the LF work is being completed. During this time, the CTU waiting for the PE task can continue with using the WT. This method also effectively improves the parallelism within the frame.

The proposed parallel method for intra-frame implementation is given in Algorithms 1 and 2. Algorithm 1 generates the MT task queue and wakes up the WT. Algorithm 2 searches the WT and performs the computation work.

The FT is responsible for initializing the work, adding the first row of PE tasks to the PE queue, and waking the WT to work. The FT also monitor the computing units that can be queued in the parallel region of each frame. The WT first searches for tasks in the PE queue. If it is the first row, it decompresses the current line. If it is not the first row, after it executes the CTU task of the current row, it returns to the thread pool and waits to be awakened for the task. If the PE queue is empty, then the thread is woken up in the LF queue. This ensures that the PE task has a higher priority than the LF task so that the former is always executed first. This ensures that enough reconstructed CTUs are executed by the LF.

# Patent rights declaration(s)

**Konkuk University does not have any current or pending patent rights relating to the technology described in this contribution.**