**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

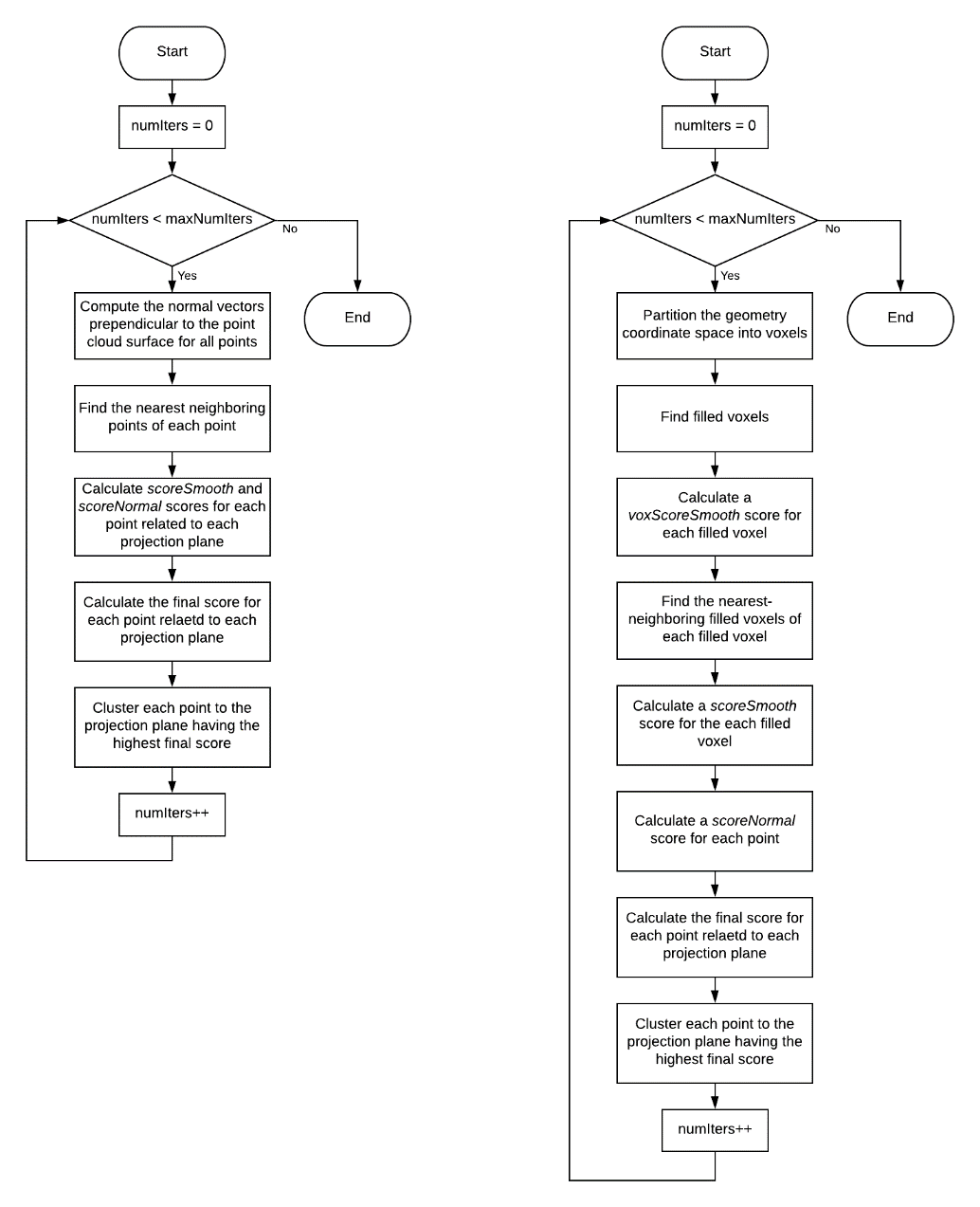
**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2019/m49587**

**July 2019, Gothenburg, Sweden**

|  |  |
| --- | --- |
| **Source** | **Samsung Electronics** |
| **Status** | **Input contribution** |
| **Title** | **[V-PCC] CE2.27 Report on Encoder’s Speedup** |
| **Author** | Esmaeil Faramarzi, Rajan Joshi, Madhukar Budagavi, Sungryeul Rhyu, and Jaeyeon Song |

# Abstract



**Figure1. Flow chart of the grid-based partitioning (GBP) method in [1]**

This contribution is a report for the core experiment CE2.27 on encoder’s speedup. In the MPEG-126 meeting, we proposed a method mainly targeting the complexity reduction of the refine segmentation procedure in TMC2 based on grid-based partitioning (GBP) of the points [1]. At the same meeting, the proposed GBP method was added to a core experiment CE2.27 for further study of its performance and complexity. This contribution reports both subjective and objective performances of Samsung’s GBP technique and analyses its complexity compared to the original refine segmentation technique in TMC2.

# Proposal

Figure 1 shows the flow chart of the BBP method proposed in [1]. It first partitions the coordinate space into a grid of voxels and finds filled voxels in the grid (i.e. voxels having at least one point inside). Then, it calculates a *voxScoreSmooth* score for each filled voxel related to each projection plane by counting the number of points in the voxel that are clustered to that projection plane through the initial segmentation process. Next, it uses KD-Tree partitioning to find the nearest-neighboring filled voxels of each filled voxel (within a search radius and limited to a maximum number of neighboring voxels). The final *scoreSmooth* for each filled voxel is then calculated by adding up the *voxScoreSmooth* values of the neighboring filled voxels. A *scoreNormal* score is also calculated for each point related to each projected plane. Next, the final score for each point related to each projection plane is calculated as the weighted linear combination of the *scoreSmooth* and *scoreNormal* scores. Finally, each point is clustered to the projection plane having the highest final score. The above steps are repeated for a few iterations.

The GBP method has the following four main parameters:

* *iterationCountRefineSegmentation*: number of iterations for the outer loop
* *voxelDimensionRefineSegmentation*: dimension of voxels in the partitioning grid (must be a power of two)
* *searchRadiusRefineSegmentation*: radius to search for the neighbouring points of each point
* *maxNNCountRefineSegmentation*: maximum number of nearest neighbouring points of each point

# Experimental results

We submitted by email the code, two sets of config files, and all-frame results related to each config set for the Samsung’s GBP method to the crosscheckers. The code, config files, and simulation spreadsheets are also available in the submission package of this report. Figure 2 shows the parameter values and the lossy simulation results related to config #1. The encoder’s self-runtime is dropped by about **80%** using config #1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | iterationCountRefineSegmentation | voxelDimensionRefineSegmentation | searchRadiusRefineSegmentation | maxNNCountRefineSegmentation |
| Loot | 10 | 4 | 192 | 1024 |
| Redandblack | 10 | 4 | 192 | 1024 |
| Soldier | 10 | 4 | 192 | 1024 |
| Queen | 10 | 4 | 192 | 1024 |
| Longdress | 50 | 4 | 192 | 1024 |
| Basketball | 20 | 4 | 192 | 1024 |
| Dancer | 20 | 4 | 192 | 1024 |





Figure 2. Parameter values and lossy simulation results of the GBP method related to config #1. The encoder’s self-runtime is dropped by about **80%**.

Figure 3 demonstrations the parameter values and lossy simulation results related to config #2. The encoder’s self-runtime is dropped by about **50%** using config #2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | iterationCountRefineSegmentation | voxelDimensionRefineSegmentation | searchRadiusRefineSegmentation | maxNNCountRefineSegmentation |
| Loot | 10 | 2 | 192 | 1024 |
| Redandblack | 10 | 2 | 192 | 1024 |
| Soldier | 10 | 2 | 192 | 1024 |
| Queen | 10 | 2 | 192 | 1024 |
| Longdress | 50 | 4 | 192 | 1024 |
| Basketball | 20 | 4 | 192 | 1024 |
| Dancer | 20 | 4 | 192 | 1024 |





Figure 3. Parameter values and lossy simulation results of the GBP method related to config #2. The encoder’s self-runtime is dropped by about **50%**.

# Visual quality

Figure 4 displays snapshots of the reconstructed point clouds using the original refine segmentation (left) and the proposed grid-based refine segmentation (right) algorithms.

Figure 5. Comparison of visual quality between the reconstructed point clouds using the original refine segmentation (left) and the proposed grid-based refine segmentation (right) algorithms.