m56594
[EE 13.50] Report on Trisoup

Kyohei Unno, Kei Kawamura

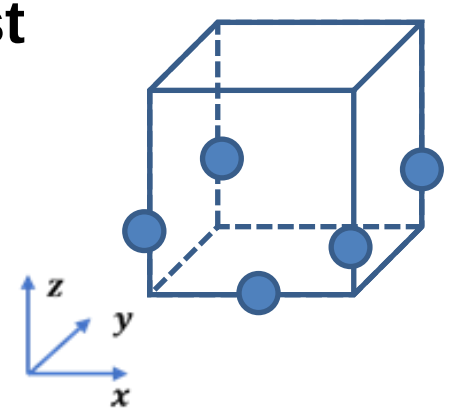
KDDI Corp. (KDDI Research, Inc.)

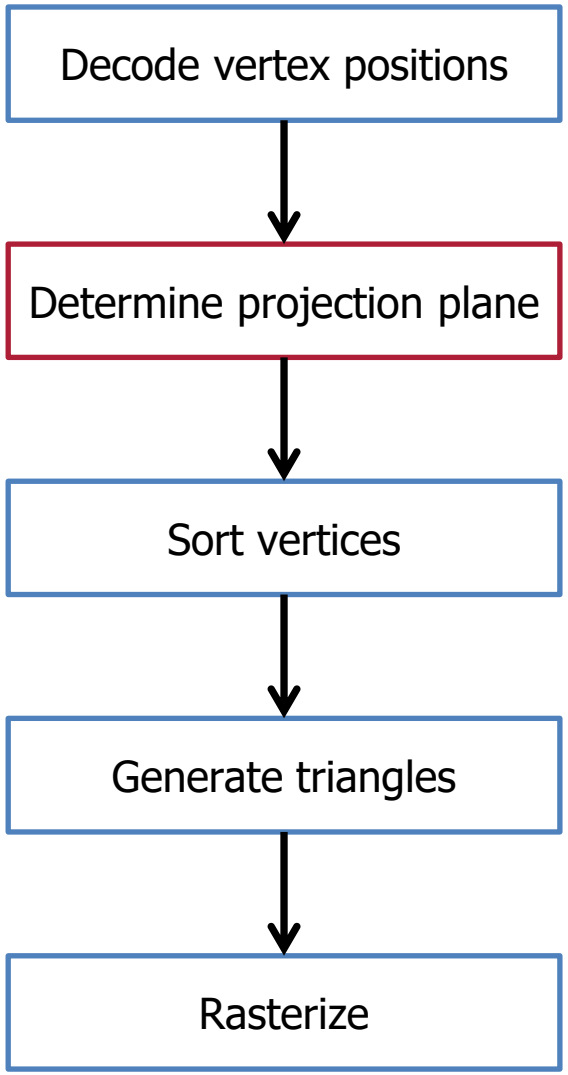
Overview

- **Goal of EE13.50**
  - **Evaluate a simplification regarding projection plane determination on Trisoup proposed in m55952.**
- **Proposal in m55952**
  - **Difference between the max. value and the min. value of vertex positions are used instead of variance to determine projection plane.**
- **Complexity Analysis (when a Trisoup node has 12 vertices (worst case))**
  - **Current implementation : (ADD, MUL, DIV, SHIFT, COMP) = (86, 36, 1, 36, 2)**
  - **Proposal in m55952 : (ADD, MUL, DIV, SHIFT, COMP) = (3, 0, 0, 0, 68)**
- **Experimental results**
  - **BD-rates for geometry are -0.1%, 0.4% (D1, D2).**
- **Related contribution of EE13.50**
  - **m56732 [G-PCC][EE13.50 related] Evaluation on Trisoup vertex sorting (KDDI)**
    Information contribution to report coding performance of two methods regarding Trisoup vertex sorting

The current implementation

- **Projection plane needs to determined for each Trisoup node.**
- **Current determining process**
  - **Calculate the variance of vertex positions for each axis.**
  - **Determine projection plane defined by axes that have lager variance (e.g. ignore an axis that has the min. variance).**
- **Variance needs square operations.**
- **It seems to be too much computation cost for the purpose.**

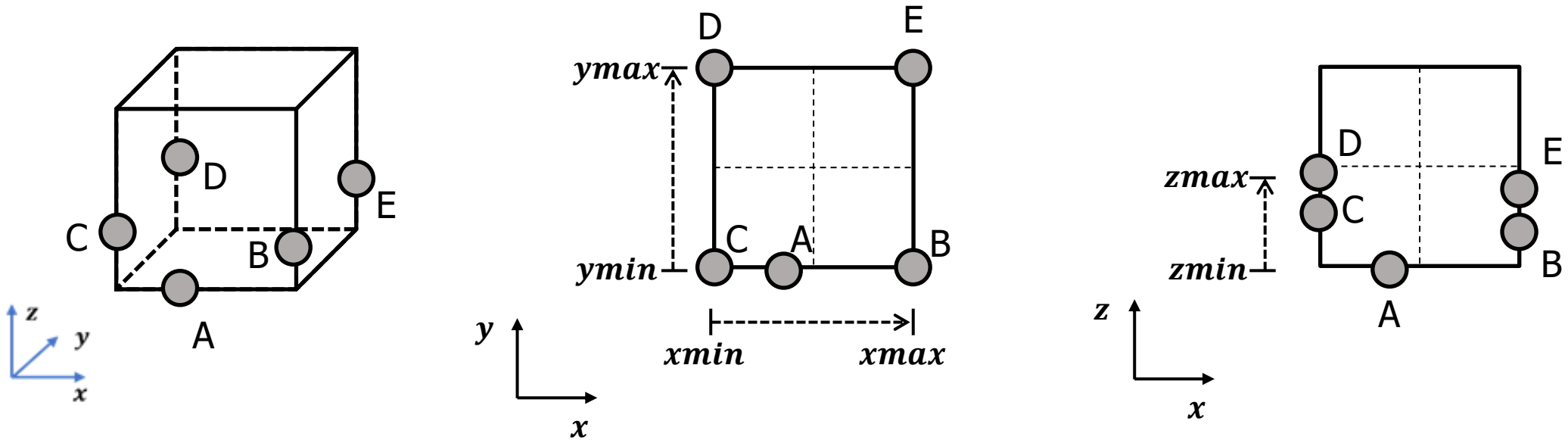




Proposal in m55952

- **Simplified method is proposed.**

- **Difference between the max. value and the min. value of vertex positions are used instead of variance.**



Complexity analysis

Worst case (a Trisoup node has 12 vertices) complexities are calculated

Current implementation

```
// Compute mean of leaf vertices. [step 1]
Vec3<int32_t> blockCentroid = 0;
for (int j = 0; j < leafVertices.size(); j++) {
  blockCentroid += leafVertices[j].pos; //add: 12
}
blockCentroid /= (int32_t)leafVertices.size(); //div: 1

// Compute variance of each component of leaf vertices. [step 2]
Vec3<int32_t> SS = 0;
for (int j = 0; j < leafVertices.size(); j++) {
  Vec3<int32_t> S = leafVertices[j].pos - blockCentroid; //add: 36
  SS += times(S, S) >> kTrisoupFpBits; //add: 36, mul: 36, shift: 36
}

// Dominant axis is the coordinate minimizing the variance. [step 3]
int32_t minSS = SS[0];
int32_t dominantAxis = 0;
for (int32_t j = 1; j < 3; j++) {
  if (minSS > SS[j]) { //comp: 2
    minSS = SS[j];
    dominantAxis = j;
  }
}
```

	ADD	MUL	DIV	SHIFT	COMP
STEP 1	12	0	1	0	0
STEP 2	72	36	0	36	0
STEP 3	0	0	0	0	2
TOTAL	84	36	1	36	2

Proposal in m55952

```
// Alternative method to determin dominant axis. [step 1]
Vec3<int32_t> min_pos = leafVertices[0].pos;
Vec3<int32_t> max_pos = leafVertices[0].pos;

for (int j = 1; j < leafVertices.size(); j++) {
    for (int axis_id = 0; axis_id < 3; axis_id++) { //comp: 66
        if (leafVertices[j].pos[axis_id] > max_pos[axis_id]){
            max_pos[axis_id] = leafVertices[j].pos[axis_id];
        } else if (leafVertices[j].pos[axis_id] < min_pos[axis_id]){
            min_pos[axis_id] = leafVertices[j].pos[axis_id];
        }
    }
}

//[step 2]
Vec3<int32_t> diff_max_min = max_pos - min_pos; //add: 3
int32_t min_diff = diff_max_min[0];
int32_t dominantAxis = 0;
for (int axis_id = 1; axis_id < 3; axis_id++) {
    if (diff_max_min[axis_id] < min_diff){ //comp: 2
        min_diff = diff_max_min[axis_id];
        dominantAxis = axis_id;
    }
}
```

	ADD	MUL	DIV	SHIFT	COMP
STEP 1	0	0	0	0	66
STEP 2	3	0	0	0	2
TOTAL	3	0	0	0	68

Experimental results

■ Conditions

- Anchor : TMC13-v12.0 + m55951 (a bugfix of Trisoup vertex sorting)
- Test : TMC13-v12.0 + m55951 + m55952
- Trisoup – RAHT (Only C2 condition, Cat1 Sequences)

■ Results

- BD-rates for geometry are -0.1%, 0.4% (D1, D2).
- Impact of the proposed simplification is very minor.
- Thank SONY for cross-checking

C2_ai	lossy geometry, lossy attributes [all intra]					
	End-to-End BD-AttrRate [%]				Geom. BD-TotGeomRate [%]	
	Luma	Chroma Cb	Chroma Cr	Reflectance	D1	D2
Cat1-A average	0.2%	0.0%	0.0%		-0.2%	0.3%
Cat1-B average	0.2%	0.4%	0.5%		0.0%	0.5%
Cat3-fused average	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
Cat3-frame average				#DIV/0!	#DIV/0!	#DIV/0!
Overall average	0.2%	0.2%	0.2%	#DIV/0!	-0.1%	0.4%
Avg. Enc Time [%]	100%					
Avg. Dec Time [%]	100%					

Additional results – Runtime comparison

KDDI Research

Conditions

- **Anchor : TMC13-v12.0 + m55951 (a bugfix of Trisoup vertex sorting)**
- **Test : TMC13-v12.0 + m55951 + m55952**
- **Trisoup – RAHT (Only C2 condition, Cat1 Sequences)**
- **Decoding runtime of projection plane determination processes are measured.**
  - Numbers of Trisoup vertices and their positions are completely same both the anchor and the test.
    - No. of vertices and positions are only depends on input point cloud and Trisoup node size.
    - Example in codec description

      “The position of a detected vertex along an edge is the average position along the edge of all such voxels adjacent to the edge among all blocks that share the edge.”
  - Encoding process and decoding process are completely same.

Results

- **The runtime of the test method is smaller 12.5% than the anchor.**
- **For all sequences and variants, the runtime of the test method always smaller than the anchor.**

Anchor		Test		Test / Anchor	
avg dec time [ms]	Total # of vertices	avg dec time [ms]	Total # of vertices	dec time	Diff. # of vertices
520.98	232367411	456.10	232367411	87.5%	0

Conclusion

- **Goal of EE13.50**
  - Evaluate a simplification regarding projection plane determination on Trisoup proposed in m55952.
- **Proposal in m55952**
  - Difference between the max. value and the min. value of vertex positions are used instead of variance to determine projection plane.
- **Complexity Analysis (when a Trisoup node has 12 vertices (worst case))**
  - Current implementation : (ADD, MUL, DIV, SHIFT, COMP) = (86, 36, 1, 36, 2)
  - Proposal in m55952 : (ADD, MUL, DIV, SHIFT, COMP) = (3, 0, 0, 0, 68)
- **Experimental results**
  - BD-rates for geometry are -0.1%, 0.4% (D1, D2).
- **Recommendation**
  - The proposal in m55952 is adopted to the next version of the test model.