m57368
Extension of Trisoup to variable node size

Kyohei Unno, Kei Kawamura

KDDI Corp. (KDDI Research, Inc.)

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.

Overview

■ Problem statement

- Current Trisoup can choose only one node size for a slice.
- It was mentioned in the previous meeting that variable node size is desired.

■ Proposed Methods

- maxTrisoupNodesize and minTrisoupNodesize are defined for each slice.
- Node size is determined by comparing parent with child cost values (currently D1 MSE is used) recursively from bottom to top.

■ Experimental results

- Three different settings for node size ranges were conducted (Test A, B, and C)
- The proposed method has gains for D1 and losses for D2 (due to the cost function).

Test	D1 BD-rate	D2 BD-rate
Test A	-2.1%	6.8%
Test B	-1.6%	1.8%
Test C	-2.5%	8.7%

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.

Background

■ Problem statement

- **Current Trisoup can choose only one node size for a slice.**
- **It was mentioned in the previous meeting that variable node size is desired.**

■ Proposal

- **An extension of Trisoup to variable node size.**

■ Related work

Variable node size Trisoup (based on TMC13-v4) was proposed in the following paper.

[2] A. Dricot, J. Ascenso, “Adaptive Multi-Level Triangle Soup for Geometry-based Point Cloud Coding,“ 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), September 2019.

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.

Over view of current method (fixed node size)

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.

Over view of proposed method (variable node size)

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.

Proposed node size determination (encoder)

1. Point clouds are reconstructed by single-level Trisoup for each depth, respectively.

2. Costs of a parent node and child nodes are compared and retain lower cost nodes recursively from bottom to top.

In [2], two variants of the cost function are proposed.
(1) D1 MSE (2) D1 + D2
This proposal only use D1 (due to time limitation...).

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.

Proposed decoding process

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.

Syntax changes

- **GPS**
  - **trisoup_multilevel_enabled_flag is introduced.**
    - When this flag is false, low-level part is the same as the current (single-level) Trisoup
- **GDUH**
  - **log2_trisoup_max_node_size_minus2 and log2_trisoup_min_node_size_minus2 are signalled.**
  - **trisoup_sampling_value_minus1 is kept single value.**
    - It may be better for preserving subjective quality consistency.

Geometry Parameter Set

geometry_parameter_set( ) {	**Descriptor**
gps_geom_parameter_set_id	ue(v)
…	
trisoup_enabled_flag	u(1)
if (trisoup_enabled_flag) {	
trisoup_multilevel_enabled_flag	u(1)
}	
…	
}	

Geometry Data Unit Header

geometry_data_unit_header( ) {	**Descriptor**
gsh_geometry_parameter_set_id	ue(v)
…	
if (trisoup_enabled_flag) {	
if(trisoup_multilevel_enabled_flag) {	
log2_trisoup_max_node_size_minus2	ue(v)
log2_trisoup_min_node_size_minus2	ue(v)
trisoup_depth = log2_trisoup_max_node_size_minus2 - log2_trisoup_min_node_size_minus2 + 1	
}else{	
log2_trisoup_node_size_minus2	ue(v)
trisoup_depth = 1	
}	
trisoup_sampling_value_minus1	ue(v)
for (i = 0; i < trisoup_depth; i++){	
unique_segments_exist_flag[i]	u(1)
if (unique_segments_exist_flag[i]) {	
num_unique_segments_bits_minus1[i]	ue(v)
num_unique_segments_minus1 [i]	u(v)
}	
}	
}	
…	
}	

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.

Experiments

Conditions

- **Anchor : TMC13-v13.0**
- **Test : TMC13-v13.0 + Proposed method**
- **Trisoup – RAHT (Only C2 condition, Cat1 Sequences)**

Node size settings

- **Three variants are tested.**
- **For Test B and Test C, the proposed method is only applied for higher rate points.**
  - Because the proposed method may less efficient for lower rate points.
  - The benefit of the proposed method is saving geometry bits by using larger node size with preserving objective quality (by cost function).

Method	Node size log2			
	r01	r02	r03	r04
Anchor (CTC)	5	4	3	2
Test A (node size = CTC & CTC+1)	6, 5	5, 4	4, 3	3, 2
Test B (max node size = 4)	5	4	4, 3	4, 3, 2
Test C (max node size = 5)	5	5, 4	5, 4, 3	5, 4, 3, 2

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.

Experiments

■ Results

- **For all test, D1 BD-rates are improved, and D2 BD-rates are degraded.**
  - Currently, only D1 is used as the cost function.
  - In [2], it is reported that D2 BD-rates were improved when D1+D2 is used as the cost function.
- **Test C has the largest gain for D1 BD-rates.**
- **Decoding time is almost same as the anchor.**

Method	Node size log2				Results			
	r01	r02	r03	r04	D1 BDR	D2 BDR	EncT	DecT
Anchor (CTC)	5	4	3	2	-	-	-	-
Test A (node size = CTC & CTC+1)	6, 5	5, 4	4, 3	3, 2	-2.1%	6.8%	173%	103%
Test B (max node size = 4)	5	4	4, 3	4, 3, 2	-1.6%	1.8%	137%	99%
Test C (max node size = 5)	5	5, 4	5, 4, 3	5, 4, 3, 2	-2.5%	8.7%	172%	99%

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.

Conclusion

- **Problem statement**
  - Current Trisoup can choose only one node size for a slice.
  - It was mentioned in the previous meeting that variable node size is desired.
- **Proposed Methods**
  - maxTrisoupNodesize and minTrisoupNodesize are defined for each slice.
  - Node size is determined by comparing parent with child cost values (currently D1 MSE is used) recursively from bottom to top.
- **Experimental results**
  - Three different settings for node size ranges were conducted (Test A, B, and C)
  - The proposed method has gains for D1 and losses for D2 (due to the cost function).

Test	D1 BD-rate	D2 BD-rate
Test A	-2.1%	6.8%
Test B	-1.6%	1.8%
Test C	-2.5%	8.7%

- **Recommendation**
  - The proposed method is study in EE (reopen EE 13.50 on Trisoup).
  - Another cost function (D1 + D2) is also studied in EE.

Thank Sony for cross-checking.

Copyright(C) 2021 KDDI Research, Inc. All Rights Reserved.