

m60214

[V-DMC][EE4.1 related][new] On improving mesh motion coding with skip mode of motion prediction

Jianfeng Xu, Kei Kawamura
KDDI Corp. (KDDI Research, Inc.)

■ Problem statement and Analysis

- Focus on mesh **motion coding** in inter frames.
- **Fact #1: In the decoded reference base mesh, there are duplicated vertices.**
 - We can search them with **no side information** at the decoder.
- **Fact #2: Motion vectors (MVs) of duplicated vertices are exactly same in most cases.**
 - It is possible to skip MV coding
- **Problem: MVs of duplicated vertices are coded as those of non-duplicated vertices.**

■ Proposed method

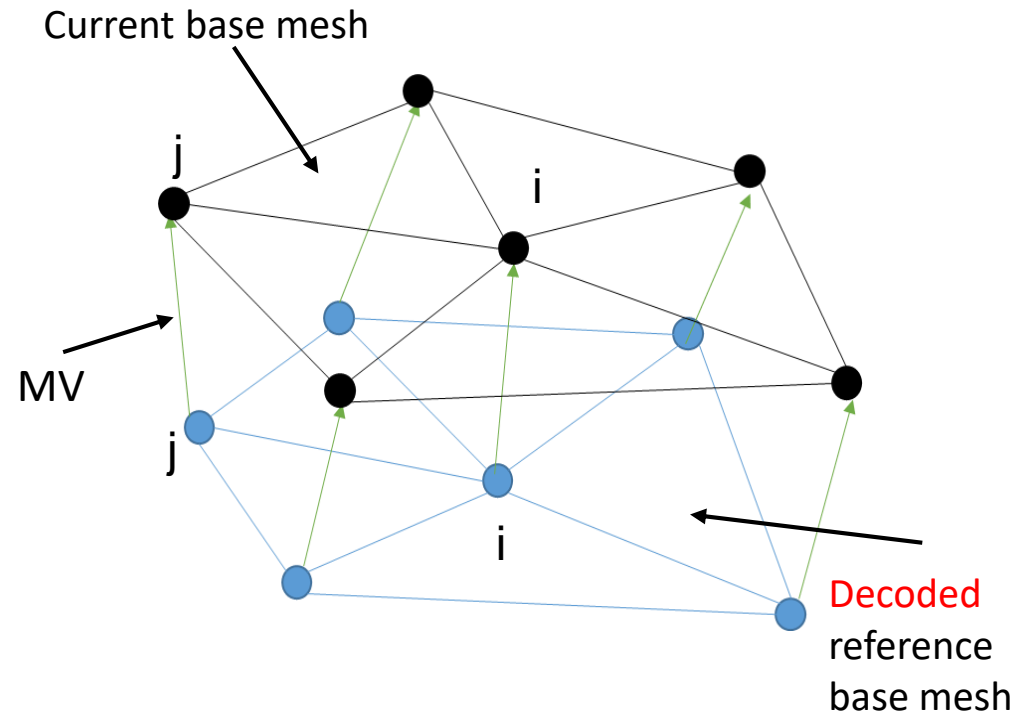
- **Encode MVs of duplicated vertices in a skip mode**
 - **Skip coding of MVs** if they are same (Most cases, no side information)
 - Code MVs if they are different (Just a few cases, need side information)

■ Simulation results

- **Decoded MVs are lossless, thus exactly same as anchor**
- **MV bits are reduced**
 - The reduction depends on the number of duplicated vertices. Maximally 20.5% (soldier_r1).

Problem statement (1)

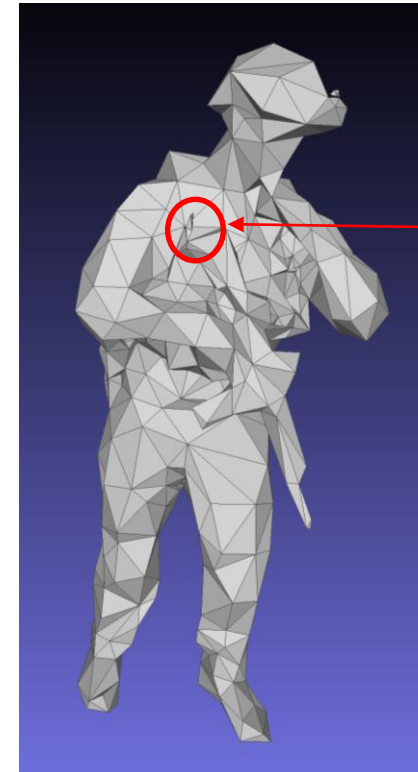
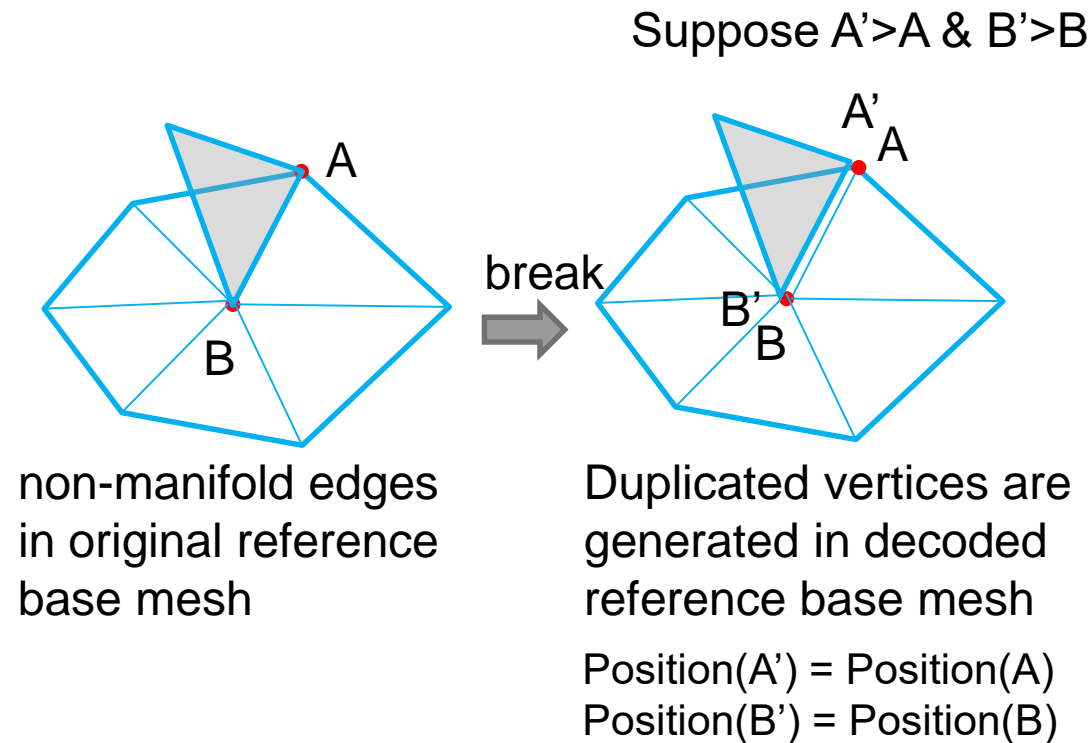
- We focus on motion coding in inter frames.
- For inter frames, MVs are coded instead of positions.
 - Decoded reference base mesh is available.
 - **1-to-1 mapping** between current base mesh and decoded reference base mesh.
 - Share vertex indexes, ordered by the traversal order.



Problem statement (2)

■ Fact #1: In the decoded reference base mesh, there are **duplicate vertices**.

- Draco breaks the non-manifold edges and generates new vertices and edges.
- Can be searched without side information at the decoder



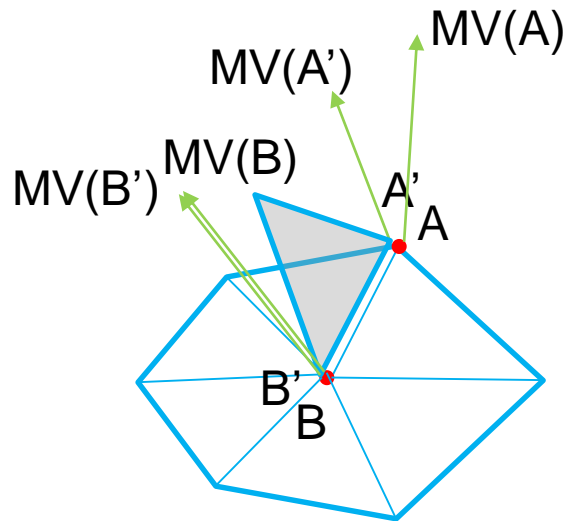
One of duplicated
vertex areas

About 15% of
vertices are
duplicated in
soldier_r1.

Soldier_r1_frame#536

Problem statement (3)

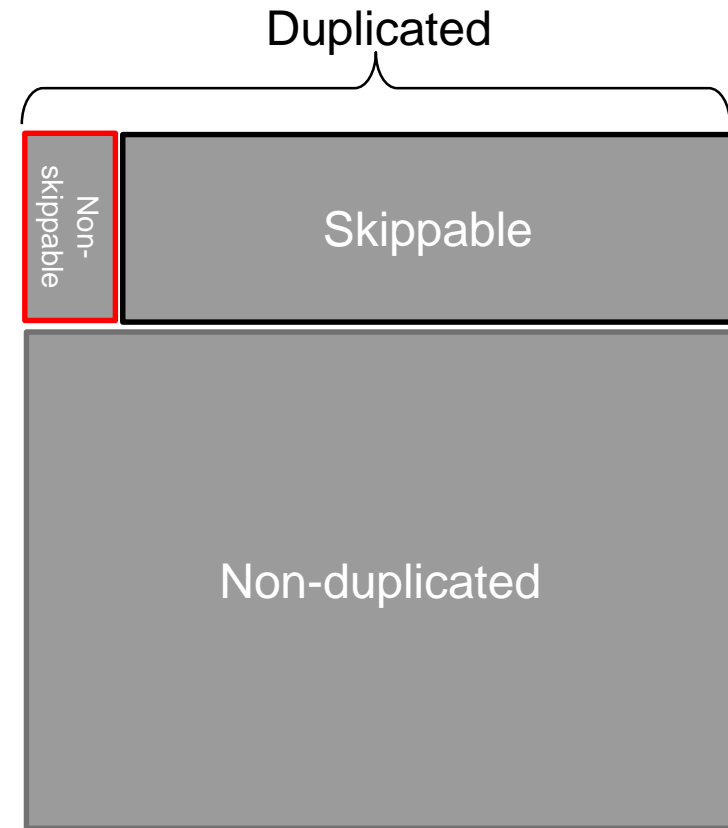
- **Fact #2: The MVs of duplicated vertices are exactly same in most cases.**
 - Only a few MVs of duplicated vertices are different.
- **The problem:**
 - MVs of duplicated vertices are coded as those of non-duplicated vertices.



MVs of duplicated vertices
are the same in most cases

Non-skippable MV: $MV(A') \neq MV(A)$
Skippable MV: $MV(B) = MV(B')$

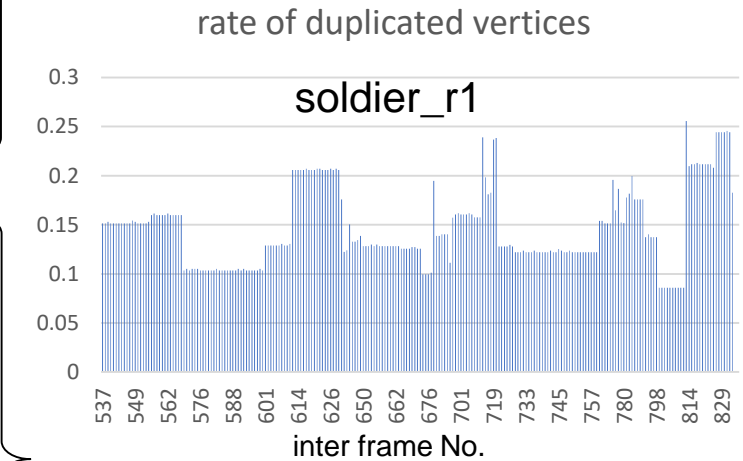
Only about 0.3% of
duplicated vertices
have non-skippable
MVs in soldier_r1.



Problem statement (4)

sequence_bitrate	average vertex number	non-duplicated vertices	duplicated vertices	rate of duplicated vertices	non-skippable MVs	skippable MVs	rate of non-skippable MVs
soldier_r1	652.58	556.30	96.28	14.75%	0.28	96.00	0.29%
soldier_r2	1259.01	1146.75	112.25	8.92%	0.32	111.93	0.28%
soldier_r3	1259.01	1146.75	112.25	8.92%	0.32	111.93	0.28%
soldier_r4	1259.01	1146.75	112.25	8.92%	0.32	111.93	0.28%
soldier_r5	1259.01	1146.75	112.25	8.92%	0.32	111.93	0.28%
levi_r1	649.96	590.57	59.39	9.14%	0.09	59.30	0.15%
levi_r2	2445.88	2392.12	53.76	2.20%	0.39	53.36	0.73%
levi_r3	2445.88	2392.12	53.76	2.20%	0.39	53.36	0.73%
levi_r4	4843.25	4791.46	51.79	1.07%	0.13	51.67	0.24%
levi_r5	-	-	-	-	-	-	-
mitch_r1	906.17	900.04	6.13	0.68%	0.09	6.04	1.46%
mitch_r2	906.17	900.04	6.13	0.68%	0.09	6.04	1.46%
mitch_r3	906.17	900.04	6.13	0.68%	0.09	6.04	1.46%
mitch_r4	14962.84	14955.80	7.05	0.05%	3.53	3.52	50.04%
mitch_r5	14962.84	14955.80	7.05	0.05%	3.53	3.52	50.04%
thomas_r1	902.21	896.98	5.22	0.58%	0.01	5.22	0.10%
thomas_r2	902.21	896.98	5.22	0.58%	0.01	5.22	0.10%
thomas_r3	902.21	896.98	5.22	0.58%	0.01	5.22	0.10%
thomas_r4	14981.64	14979.06	2.58	0.02%	1.39	1.19	54.01%
thomas_r5	14981.64	14979.06	2.58	0.02%	1.39	1.19	54.01%

There are many duplicated vertices and most of them have skippable MVs.

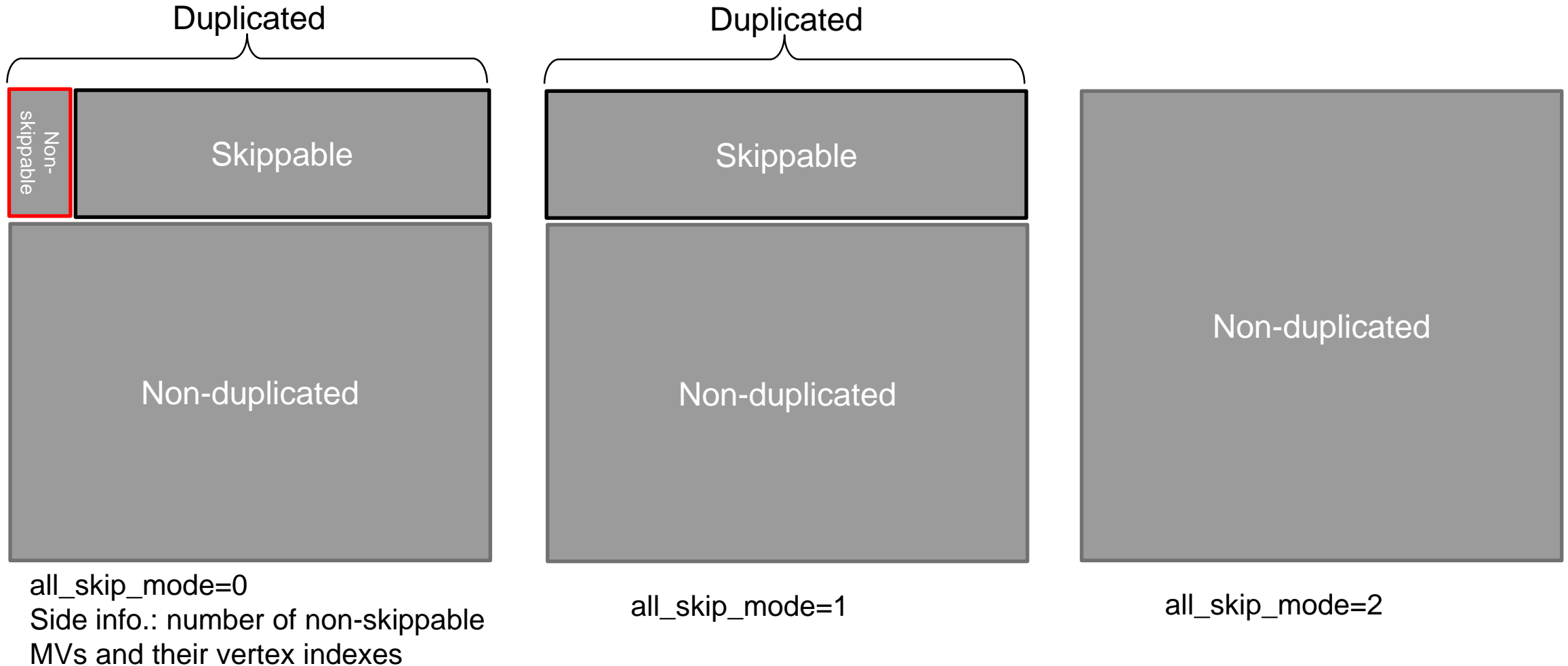


Not so many duplicated vertices

Proposed method

■ Encode MVs of duplicated vertices in a **skip mode**

- **Skip coding of MVs** if they are same (Most cases, no side information)
- **Code MVs** if they are different (Just a few cases, need side information)



Syntax (and decoding process)

	Descriptor
if(vuh_mesh_data_motion_field_present_flag) { // decode MVs in an inter frame	
all_skip_mode	u(2)
if(all_skip_mode==0) { // duplicated vertices are available and non-skippable MVs are available	
non-skippable_vertex_num	ue(v)
for(v=0; v< non-skippable_vertex_num; v++){	
non-skippable_vertex_index[v]	ue(v)
}	
for(v=0; v<vertex_num; v++){	
if(v ∈ duplicated_vertex_index && v ∉ non-skippable_vertex_index) {	
MV[v] = MV[corresponding_vertex[v]]	
}	
else { // decode MV for non-duplicated vertex or non-skippable vertex	
MV[v] = DecodeMV(v)	
}	
}	
}	
else if(all_skip_mode==1) { // duplicated vertices are available and non-skippable MVs are unavailable	
for(v=0; v<vertex_num; v++){	
if(v ∈ duplicated_vertex_index) { // copy MV for duplicated vertex	
MV[v] = MV[corresponding_vertex_index[v]]	
}	
else { // decode MV for non-duplicated vertex	
MV[v] = DecodeMV(v)	
}	
}	
}	
else if(all_skip_mode==2) { // not use skip_mode	
for(v=0; v<vertex_num; v++){ // decode MV for each vertex in base mesh	
MV[v] = DecodeMV(v)	
}	
}	

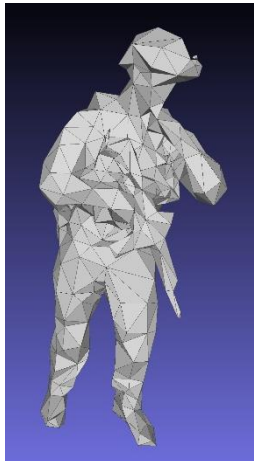
Simulation results (1)

■ Experimental condition

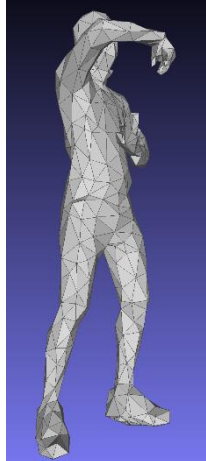
- Ubuntu 18.04, clang-13, libc++-13
- Configure: same as anchor
- Test sequences: all four sequences with inter frames in the above configures
 - levi, soldier, mitch, and thomas

■ Evaluations

- BD-rate
- Motion bit amount
- Comparison with anchor (reproduced by KDDI, the same results as [m60254])



Soldier_r1



Levi_r1



Mitch_r1



Thomas_r1

Simulation results (2)

	anchor	proposal			Comparison	
Sequence_bitrate	total motion bits	MV bits	side info.	total motion bits	saved bits	bits ratio
soldier_r1	737,176	583,512	2,384	585,896	151,280	79%
soldier_r2	790,176	666,448	1,760	668,208	121,968	85%
soldier_r3	790,176	666,448	1,760	668,208	121,968	85%
soldier_r4	790,176	666,448	1,760	668,208	121,968	85%
soldier_r5	790,176	666,448	1,760	668,208	121,968	85%
levi_r1	146,464	132,520	200	132,720	13,744	91%
levi_r2	552,048	535,488	368	535,856	16,192	97%
levi_r3	552,048	535,488	368	535,856	16,192	97%
levi_r4	668,200	657,096	216	657,312	10,888	98%
levi_r5	-	-	-	-	-	-
mitch_r1	868,896	862,440	1,656	864,096	4,800	99.45%
mitch_r2	868,896	862,440	1,656	864,096	4,800	99.45%
mitch_r3	868,896	862,440	1,656	864,096	4,800	99.45%
mitch_r4	12,030,848	12,028,488	7,240	12,035,728	-4,880	100.04%
mitch_r5	12,030,848	12,028,488	7,240	12,035,728	-4,880	100.04%
thomas_r1	674,832	670,968	1,552	672,520	2,312	99.66%
thomas_r2	674,832	670,968	1,552	672,520	2,312	99.66%
thomas_r3	674,832	670,968	1,552	672,520	2,312	99.66%
thomas_r4	10,726,008	10,725,328	3,792	10,729,120	-3,112	100.03%
thomas_r5	10,726,008	10,725,328	3,792	10,729,120	-3,112	100.03%

Averagely **11%** of MV bits are reduced
without any degradation of decoded meshes

Because the number of duplicated vertices is rather small, the gain is rather small too.

Simulation results (3)

Inter, Random Access																
lossy geometry, lossy attributes [inter, random access]																
C2_ra	Pointcloud-based BD-TotalRate [%]					Image-based BD-TotalRate [%]		Pointcloud-based BD-TotGeomRate [%]		Pointcloud-based BD-AttrRate [%]			Image-based BD-TotGeomRate [%]		Image-based BD-AttrRate [%]	
	D1	D2	Luma	Chroma_Cb	Chroma_Cr	Geom	Luma	D1	D2	Luma	Chroma_Cb	Chroma_Cr	Geom	Luma	Chroma_Cb	Chroma_Cr
Cat1-A Average	-0.3%	-0.2%	-0.2%	-0.1%	-0.1%	-0.3%	-0.2%	-1.1%	-1.0%	0.0%	0.0%	0.0%	-1.1%	-1.1%	0.0%	0.0%
Cat1-B Average	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
Cat1-C Average	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	#VALUE!	#VALUE!	0.0%	0.0%	0.0%	#VALUE!	#VALUE!	0.0%	0.0%
Overall average	-0.1%	-0.1%	-0.1%	0.0%	0.0%	-0.1%	-0.1%	0%	0%	0.0%	0.0%	0.0%	0%	0%	0.0%	0.0%
Avg. Enc Time [%]																
Avg. Dec Time [%]																

		anchor (reproduced by KDDI)																				
Sequence	Variant	Rate	Rate	Rate	Rate	Total Bits	No. output	Bitstream [bits]				Pointcloud-based PSNR [dB]					Image-based PSNR [dB]		Geometry Sub-bitstreams [bits]			
		(Atlas)	(Geom)	(Texture)	(Atlas)	per input		Total	Geometry	Texture	Atlas	D1	D2	Luma	Chroma Cb	Chroma Cr	Geom	Luma	Base Mesh	Displacements	Motion	
		[Mbps]	[Mbps]	[Mbps]	[Mbps]	face [bpif]	faces															
soldier_voxelised	r01	3.889	0.647	0.646	3.242	0.001	4.32	22469888	38892856	6,461,584	32423168	8,104	63.69	64.79	33.36	45.68	46.07	43.72	29.32	2146232	3578176	737176
soldier_voxelised	r02	7.922	2.530	2.530	5.391	0.001	8.80	45323328	79217528	25,296,272	53913680	7,576	70.41	72.44	35.16	46.01	46.46	47.55	32.64	6655376	17850720	790176
soldier_voxelised	r03	10.474	4.291	4.290	6.183	0.001	11.64	45323328	104740320	42,903,024	61829720	7,576	72.27	74.71	35.65	46.38	46.86	48.57	33.53	6655376	35457472	790176
soldier_voxelised	r04	13.641	4.291	4.290	9.350	0.001	15.16	45323328	136407984	42,903,024	93497384	7,576	72.27	74.71	36.59	47.43	48.02	48.57	34.06	6655376	35457472	790176
soldier_voxelised	r05	20.828	4.756	4.755	16.072	0.001	23.14	45323328	208281592	47,554,032	160719984	7,576	72.63	75.18	37.52	48.48	49.11	48.74	34.67	6655376	40108480	790176

Sequence	Variant	proposal																						
		Rate [Mbps]	Rate (Geom+ Atlas) [Mbps]	Rate (Geom) [Mbps]	Rate (Texture) [Mbps]	Rate (Atlas) [Mbps]	Total Bits per input face [bpif]	No. output faces	Bitstream [bits]				Pointcloud-based PSNR [dB]					Image-based PSNR [dB]		Geometry Sub-bitstreams [bits]				
									Total	Total- skipMode	Geometry	Texture	Atlas	D1	D2	Luma	Chroma Cb	Chroma Cr	Geom	Luma	Base Mesh	Displacements	Motion	skipMode
soldier_voxelised	r01	3.874	0.632	0.631	3.242	0.001	4.30	22469888	38741576	38739192	6,310,304	32423168	8104	63.69	64.79	33.36	45.68	46.07	43.72	29.32	2146232	3578176	583512	2384
soldier_voxelised	r02	7.910	2.518	2.517	5.391	0.001	8.79	45323328	79095560	79093800	25,174,304	53913680	7576	70.41	72.44	35.16	46.01	46.46	47.55	32.64	6655376	17850720	666448	1760
soldier_voxelised	r03	10.462	4.279	4.278	6.183	0.001	11.62	45323328	104618352	104616592	42,781,056	61829720	7576	72.27	74.71	35.65	46.38	46.86	48.57	33.53	6655376	35457472	666448	1760
soldier_voxelised	r04	13.629	4.279	4.278	9.350	0.001	15.14	45323328	136286016	136284256	42,781,056	93497384	7576	72.27	74.71	36.59	47.43	48.02	48.57	34.06	6655376	35457472	666448	1760
soldier_voxelised	r05	20.816	4.744	4.743	16.072	0.001	23.13	45323328	208159624	208157864	47,432,064	160719984	7576	72.63	75.18	37.52	48.48	49.11	48.74	34.67	6655376	40108480	66448	1760

- The amount of MV bits is very small compared to the total bits, which leads to the total BD-rate gain is very small.
- However, the proposed method is **simple and effective**.
- Also, we hope that more inter frames will be available in the future.

■ Problem statement and Analysis

- Focus on mesh motion coding in inter frames.
- **Fact #1: In the decoded reference base mesh, there are duplicated vertices.**
 - We can search them with no side information at the decoder.
- **Fact #2: MVs of duplicated vertices are exactly same in most cases.**
 - It is possible to skip MV coding
- **Problem: MVs of duplicated vertices are coded as those of non-duplicated vertices.**

■ Proposed method

- **Encode MVs of duplicated vertices in a skip mode**
 - Skip coding of MVs if they are same (Most cases, no side information)
 - Code MVs if they are different (Just a few cases, need side information)

■ Simulation results

- **Decoded MVs are lossless, thus exactly same as anchor**
- **MV bits are reduced**
 - The reduction depends on the number of duplicated vertices. Maximally 20.5% (soldier_r1).

■ Recommendation

- **Adopt the proposal to the future V-DMC specification and reference software.**

