JCT3V-D0089: 3D-HEVC: Adaptive Virtual Depth Block Partition for View Synthesis Prediction and Complexity Analysis

National Cheng Kung University (NCKU)
Chun-Fu Chen, Gwo Giun (Chris) Lee, Bo-Syun Li

Hong Kong Applied Science and Technology Research Institute (ASTRI)
Chunhui Cui, Yan Huo

Abstract

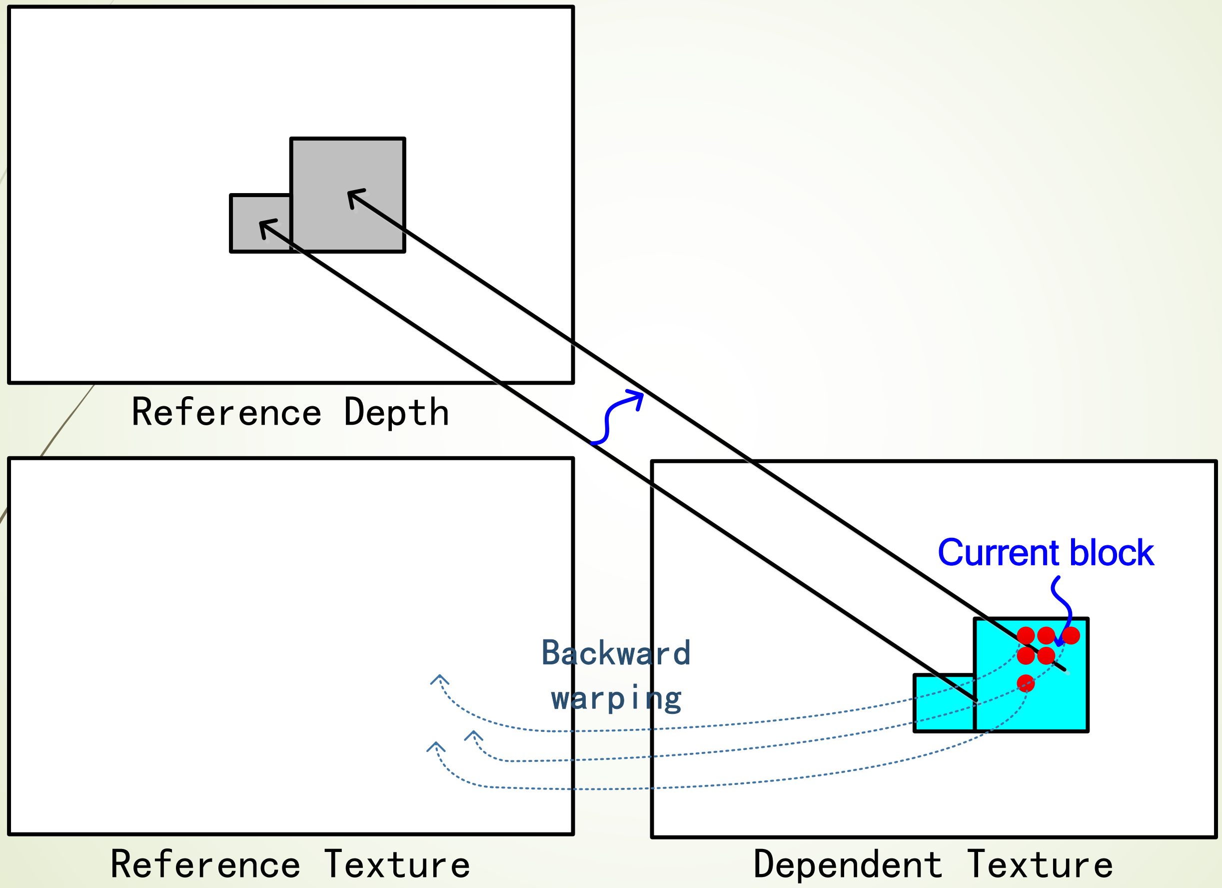
- In this proposal
  - Adaptively partition the depth block to derive the disparity vectors (DoNBDV)
  - Supported block size: 8x8, 8x4, 4x8 block
  - The complexity analysis of test 04 is provided.
- Experimental results (average)

	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	dec time
Test 01	0.0%	0.0%	0.0%	101.3%
Test 02	0.0%	0.0%	0.0%	100.2%
Test 03	0.0%	0.0%	0.0%	100.4%
Test 04	0.1%	0.0%	0.0%	100.2%

Outline

- Abstract
- Introduction to Backward View Synthesis Prediction
- Motivation
- Proposed Algorithm
- Experimental Results
- Complexity Assessment
- Conclusion

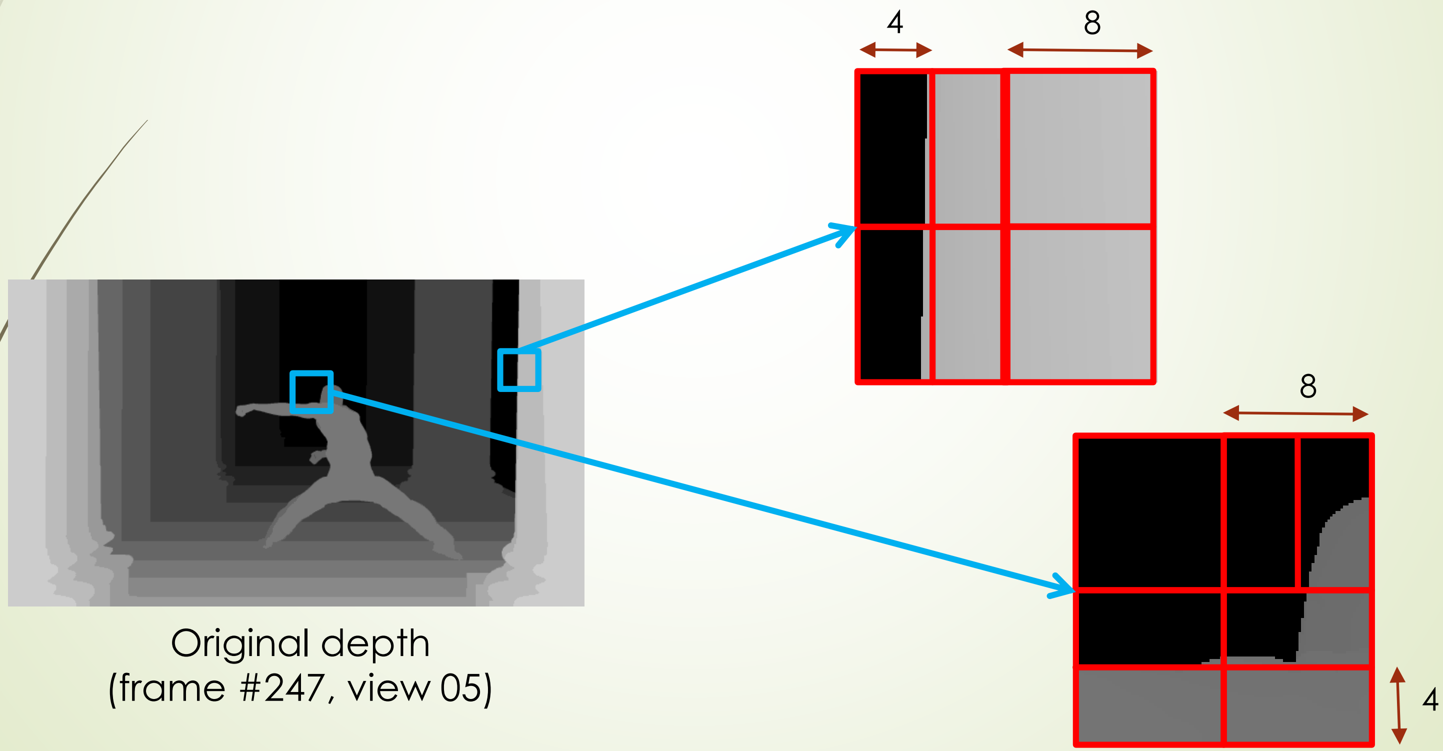
Introduction to Backward View Synthesis Prediction (B-VSP)



NBDV: Neighboring block based disparity vector
DoNBDV: Depth-oriented neighboring block based disparity vector

Motivation

- Block size becomes larger → DV may be inaccurate
- Adaptive block partition to obtain the DVs.

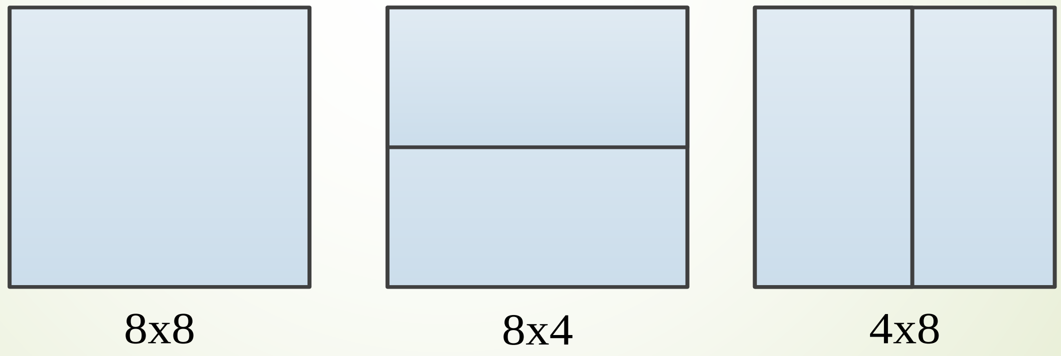


Original depth
(frame #247, view 05)

Proposed Algorithm (1/5)

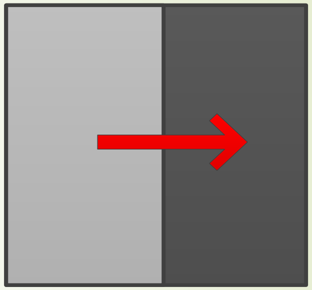
- Modify the second disparity vector derivation (DoNBDV) in BVSP
  - Operation at 8x8 block basis
  - Content adaptively partition depth block → three block types

Three depth block partition types

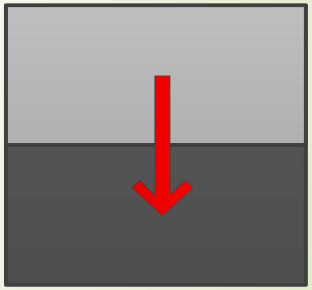


Proposed Algorithm (2/5)

- Content adaptive depth block partition
  - According to the gradient of a 8x8 depth block
  - Horizontal depth transition detection
    - $NoLC_H = \sum_{k=0}^{N} LineChange[k]$
    - $LineChange[k] = \begin{cases} 1, Grad_H[k] > TH_H \\ 0, \quad Otherwise \end{cases}$
  - Vertical depth transition detection
    - $NoLC_V = \sum_{k=0}^{N} LineChange[k]$
    - $LineChange[k] = \begin{cases} 1, Grad_V[k] > TH_V \\ 0, \quad Otherwise \end{cases}$



4x8



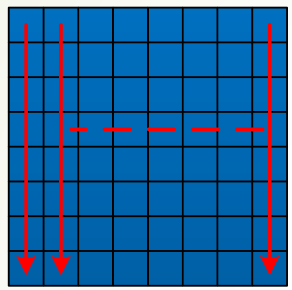
8x4

$Grad_H$: Horizontal gradient
$Grad_V$: Vertical gradient
$NoDC_H$: Number of depth changes in horizontal
$NoDC_V$: Number of depth changes in vertical

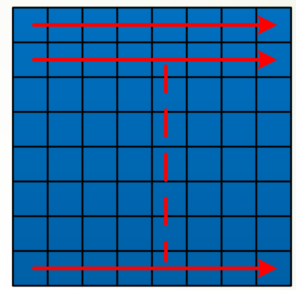
Proposed Algorithm (3/5)

- Depth block analysis

Method 1: Full-pixels gradient computation

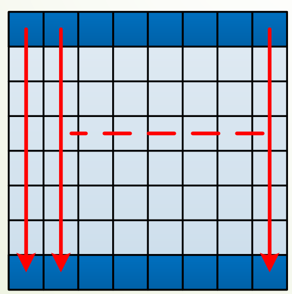


Vertical Direction

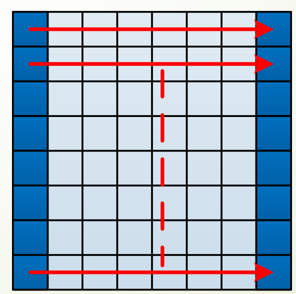


Horizontal Direction

Method 2: Outmost pixels gradient computation



Vertical Direction

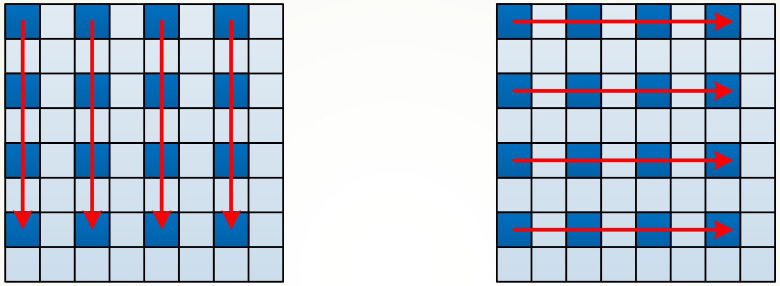


Horizontal Direction

Proposed Algorithm (4/5)

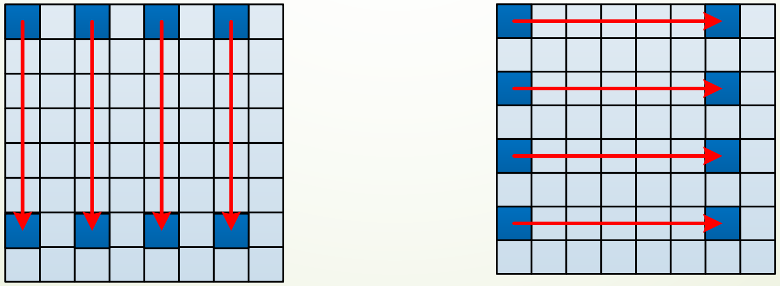
- Depth block analysis

Method 3: Full- pixels gradient computation with down-sampling by 2



Vertical Direction Horizontal Direction

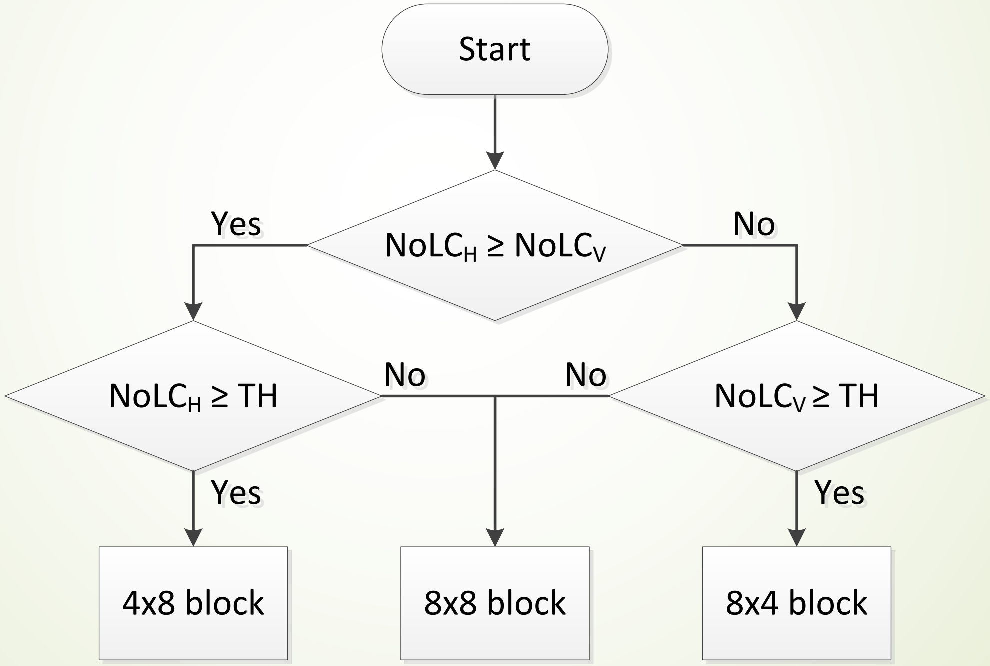
Method 4: Outmost pixels gradient computation with down-sampling by 2



Vertical Direction Horizontal Direction

Proposed Algorithm (5/5)

- Content adaptive depth block partition
  - According to the number of line changes in horizontal and vertical to determine partition type



$NoLC_H$: Number of line changes in horizontal direction
$NoLC_V$: Number of line changes in vertical direction

Experimental Results (1/4)

- Method 01: Full pixels gradient computation, TH =4

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time	ren time
Balloons	0.0%	-0.1%	0.0%	0.0%	0.0%	-0.1%	100.7%	100.1%	100.0%
Kendo	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	100.6%	100.3%	99.9%
Newspaper_CC	0.0%	-0.2%	0.1%	0.0%	0.0%	0.0%	100.7%	99.8%	100.0%
GT_Fly	0.0%	0.5%	0.1%	0.1%	0.1%	0.1%	100.6%	105.3%	99.9%
Poznan_Hall2	0.0%	-0.5%	-0.3%	-0.2%	-0.2%	-0.1%	100.6%	100.8%	99.8%
Poznan_Street	0.0%	0.3%	0.2%	0.1%	0.1%	0.1%	100.2%	100.3%	101.7%
Undo_Dancer	0.0%	0.6%	0.7%	0.2%	0.1%	-0.2%	100.5%	102.4%	100.2%
1024x768	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	100.6%	100.1%	100.0%
1920x1088	0.0%	0.2%	0.2%	0.0%	0.0%	-0.1%	100.5%	102.2%	100.4%
average	**0.0%**	**0.1%**	**0.1%**	**0.0%**	**0.0%**	**0.0%**	**100.5%**	**101.3%**	**100.2%**

Experimental Results (2/4)

- Method 02: Outmost pixels gradient computation, TH =4

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time	ren time
Balloons	0.0%	0.0%	0.0%	0.0%	0.0%	-0.1%	100.2%	100.2%	101.1%
Kendo	0.0%	0.2%	0.1%	0.1%	0.0%	0.1%	100.5%	99.8%	99.8%
Newspaper_CC	0.0%	-0.1%	0.1%	0.0%	0.0%	0.0%	100.3%	99.2%	100.3%
GT_Fly	0.0%	0.4%	0.2%	0.1%	0.1%	0.0%	100.3%	102.0%	100.6%
Poznan_Hall2	0.0%	-0.5%	-0.1%	-0.1%	-0.1%	-0.2%	100.1%	99.9%	99.8%
Poznan_Street	0.0%	0.4%	0.3%	0.1%	0.1%	0.1%	100.3%	99.6%	100.7%
Undo_Dancer	0.0%	0.7%	0.8%	0.2%	0.2%	-0.1%	100.1%	100.8%	100.3%
1024x768	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	100.3%	99.8%	100.4%
1920x1088	0.0%	0.3%	0.3%	0.1%	0.0%	0.0%	100.2%	100.6%	100.3%
average	**0.0%**	**0.2%**	**0.2%**	**0.0%**	**0.0%**	**0.0%**	**100.3%**	**100.2%**	**100.4%**

Experimental Results (3/4)

- Method 03: Full pixels gradient computation downsampled by 2,
  TH =2

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time	ren time
Balloons	0.0%	-0.1%	0.0%	0.0%	0.0%	-0.1%	100.2%	99.9%	99.9%
Kendo	0.0%	0.0%	-0.1%	0.0%	0.0%	0.0%	100.3%	100.1%	99.9%
Newspaper_CC	0.0%	-0.2%	0.0%	0.0%	0.0%	-0.1%	100.3%	99.6%	100.1%
GT_Fly	0.0%	0.4%	0.5%	0.1%	0.1%	0.1%	100.2%	102.4%	99.9%
Poznan_Hall2	0.0%	-0.3%	0.0%	-0.1%	-0.1%	-0.1%	100.0%	100.1%	99.6%
Poznan_Street	0.0%	0.3%	0.4%	0.1%	0.1%	0.1%	99.2%	99.8%	100.5%
Undo_Dancer	0.0%	0.9%	0.9%	0.2%	0.2%	0.0%	100.1%	101.0%	100.4%
1024x768	0.0%	-0.1%	0.0%	0.0%	0.0%	-0.1%	100.3%	99.9%	99.9%
1920x1088	0.0%	0.3%	0.5%	0.1%	0.1%	0.0%	99.9%	100.9%	100.1%
average	**0.0%**	**0.2%**	**0.2%**	**0.0%**	**0.0%**	**0.0%**	**100.1%**	**100.4%**	**100.0%**

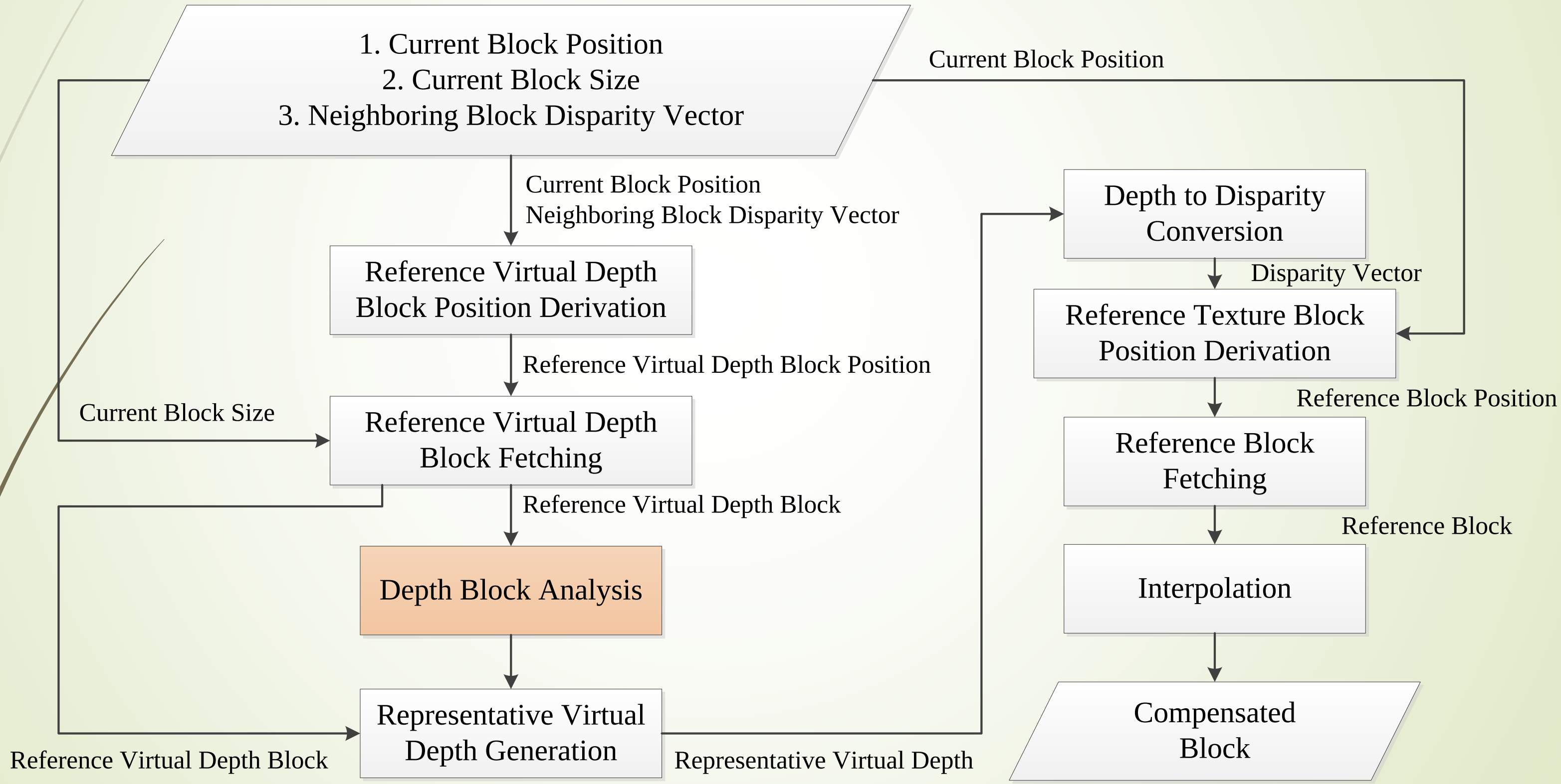
Experimental Results (4/4)

- Method 04: Outmost pixels gradient computation downsampled by 2,
  TH =2

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time	ren time
Balloons	0.0%	-0.1%	-0.1%	-0.1%	-0.1%	0.0%	100.1%	99.9%	100.3%
Kendo	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.3%	100.1%	99.5%
Newspaper_CC	0.0%	-0.1%	0.1%	0.0%	0.0%	0.0%	100.1%	99.7%	99.7%
GT_Fly	0.0%	0.7%	0.3%	0.1%	0.1%	0.1%	100.2%	101.6%	100.1%
Poznan_Hall2	0.0%	0.0%	0.2%	0.0%	0.0%	0.1%	100.1%	100.2%	100.1%
Poznan_Street	0.0%	0.5%	0.2%	0.1%	0.1%	0.1%	100.2%	99.4%	100.6%
Undo_Dancer	0.0%	0.7%	1.0%	0.2%	0.2%	0.0%	99.9%	100.5%	99.7%
1024x768	0.0%	-0.1%	0.0%	0.0%	0.0%	0.0%	100.2%	99.9%	99.8%
1920x1088	0.0%	0.5%	0.4%	0.1%	0.1%	0.1%	100.1%	100.4%	100.1%
average	**0.0%**	**0.3%**	**0.2%**	**0.1%**	**0.0%**	**0.0%**	**100.1%**	**100.2%**	**100.0%**

Complexity Assessment(1/8)

- Flowchart of proposed B-VSP



Complexity Assessment(2/8)

- Data granularity
  - One PU, including 64x64, 64x32, etc
- Major process of the proposed B-VSP
  - Depth content analysis
  - Representative depth generation
  - Interpolation

Complexity Assessment(3/8)

- Number of operations (data granularity is one PU)
  - Depth content analysis

$$Grad_V[i] = abs\left(Depth[j, 2\times i] - Depth[j+7, 2\times i]\right),\ 0 \le i \le 3$$

$$Th_V = \left(\underset{0\le i\le 3}{MAX}\left(Grad_V[i]\right)\right) >> 1$$

$$NoLC = \sum_{i=0}^{3} LineChange[i],\ LineChange[i] = \begin{cases} 1, & one\ of\ \ Grad_V[i] > TH_V \\ 0, & otherwise \end{cases}$$

  - Representative virtual depth generation

$$\mathrm{Re}\, presentativeDepth = \underset{0\le j\le subblock\_H,\ 0\le i\le subblock\_V}{MAX}\left(Depth[j,i]\right)$$

  - Interpolation

$Luma:$

$2/4\ pixel\ position:\ filter_1 = [-1,\ 4,\ -11,\ 40,\ 40,\ 5,\ -11,\ 5,\ -1]$

$Chroma:$

$1/8\ pixel\ position:\ filter_x = [-2,\ 58,\ 10,\ -2]$

Complexity Assessment(4/8)

Data storage requirement for one PU (bits)		
	Proposed B-VSP	Anchor
Reference virtual depth block	MxNx8	MxNx8
Reference texture block with extended taps	(3+8+4)x8x8	(3+4+4)x4x8
Compensated block	MxNx8	MxNx8
Analyzed gradient map and buffer of depth maximums in one 4x4 block	((4+4)+4)x8	N/A

M is the block width and N is the block height in horizontal and vertical direction respectively; the bits of one pixel of texture or depth is 8 bits.

Complexity Assessment(5/8)

Data transfer rate for one PU (bits/PU)		
	Proposed B-VSP	Anchor
Reference virtual depth block	MxNx8	MxNx8
Reference texture block with extended taps	(3+8+4)x8x(M/8)x(N/8)x8 + (1+4+2)x8x(M/4)x(N/4)x8	(3+4+4)x4x(M/4)x(N/4)x8 + (1+2+2)x8x(M/2)x(N/2)x8
Compensated block	MxNx8 + (M/2)x(N/2)x8	MxNx8 + (M/2)x(N/2)x8

M is the block width and N is the block height in horizontal and vertical direction respectively; the bits of one pixel of texture or depth is 8 bits.

Complexity Assessment(6/8)

Number of operations						
	Proposal (Average)		Anchor		Ratio (to anchor)	
PU_size	Add/Sub/ Abs/Comp	Cons_Mul	Add/Sub/ Abs/Comp	Cons_Mul	Add/Sub/ Abs/Comp	Cons_Mul
64x64	41024	32768	38912	32768	105.43%	100.00%
64x32	20512	16384	19456	16384	105.43%	100.00%
32x64	20512	16384	19456	16384	105.43%	100.00%
32x32	10256	8192	9728	8192	105.43%	100.00%
32x16	5128	4096	4864	4096	105.43%	100.00%
16x32	5128	4096	4864	4096	105.43%	100.00%
16x16	2564	2048	2432	2048	105.43%	100.00%
16x8	1282	1024	1216	1024	105.43%	100.00%
8x16	1282	1024	1216	1024	105.43%	100.00%
8x8	641	512	608	512	105.43%	100.00%
8x4	304	256	304	256	100.00%	100.00%
4x8	304	256	304	256	100.00%	100.00%
Avg.					104.52%	100.00%

Complexity Assessment(7/8)

Data storage requirement (Bits)				
PU_size	Proposal (Average)	Anchor	Ratio (to anchor)	Difference bits (to anchor)
64x64	66604	65900	101.07%	704
64x32	33836	33132	102.12%	704
32x64	33836	33132	102.12%	704
32x32	17452	16748	104.20%	704
32x16	9260	8556	108.23%	704
16x32	9260	8556	108.23%	704
16x16	5164	4460	115.78%	704
16x8	3116	2412	129.19%	704
8x16	3116	2412	129.19%	704
8x8	2092	1388	150.72%	704
8x4	1100	876	125.57%	224
4x8	1324	876	151.14%	448
Average			118.96%	643

Complexity Assessment(8/8)

Data transfer rate (Bits/PU)			
PU_size	Proposal (Average)	Anchor	Ratio (to anchor)
64x64	185698	213004	87.18%
64x32	92855	114700	80.95%
32x64	92855	114700	80.95%
32x32	46434	57356	80.96%
32x16	23223	28684	80.96%
16x32	23223	28684	80.96%
16x16	11618	14348	80.97%
16x8	5815	7180	80.99%
8x16	5815	7180	80.99%
8x8	2914	3596	81.03%
8x4	1356	1804	75.17%
4x8	1676	1804	92.90%
Average			82.00%

Conclusion

- Coding Performance of method 4
  - 0.1% bitrate increasing for video PSNR vs. video bitrate
  - Slightly bitrate increasing video PSNR vs. total bitrate and synth PSNR vs. total bitrate in average
  - 0.2 % decoding time increasing
- Complexity of method 4 vs. current B-VSP
  - About 4.5 % Number of operations increasing → Depth block analysis
  - About 19% data storage size increasing
    - Larger block size → more reference texture data
    - Exactly increased size → about 480 ~ 736 bits
  - About 18 % data transfer rate per PU decreasing
  - Number of data fetching from decoded picture buffer is reduced
- Adopted method 4 and further studied on larger sub-block.

Thanks for ITRI's crosscheck on test 02 and test 04(JCT3V-D0292)

Appendix

Complexity Assessment – Luma only

- Number of operations (data granularity is one PU)
  - Depth content analysis

$$Grad_V[i] = abs\left(Depth[j, 2\times i] - Depth[j+7, 2\times i]\right),\ 0 \le i \le 3$$

$$Th_V = \left(\underset{0\le i\le 3}{MAX}\left(Grad_V[i]\right)\right) >> 1$$

$$NoLC = \sum_{i=0}^{3} LineChange[i],\ LineChange[i] = \begin{cases} 1, & one\ of\ \ Grad_V[i] > TH_V \\ 0, & otherwise \end{cases}$$

  - Representative virtual depth generation

$$\mathrm{Re}\ presentativeDepth = \underset{0\le j\le subblock\_H,\ 0\le i\le subblock\_V}{MAX}\left(Depth[j,i]\right)$$

  - Interpolation:

$$1/4\ pixel\ position:\ filter_0 = [-1,\ 4,\ 10,\ 58,\ 17,\ -5,\ 1]$$

$$2/4\ pixel\ position:\ filter_1 = [-1,\ 4,\ -11,\ 40,\ 40,\ 5,\ -11,\ 5,\ -1]$$

$$3/4\ pixel\ position:\ filter_2 = [1,\ -5,\ 17,\ 58,\ 10,\ 4,\ -1]$$

Complexity Assessment – Luma only

Data storage requirement for one PU (bits)		
	Proposed B-VSP	Anchor
Reference virtual depth block	MxNx8	MxNx8
Reference texture block with extended taps	(3+8+4)x8x8	(3+4+4)x4x8
Compensated block	MxNx8	MxNx8
Analyzed gradient map and buffer of depth maximums in one 4x4 block	((4+4)+4)x8	N/A

M is the block width and N is the block height in horizontal and vertical direction respectively; the bits of one pixel of texture or depth is 8 bits.

Complexity Assessment – Luma only

Data transfer rate for one PU (bits/PU)		
	Proposed B-VSP	Anchor
Reference virtual depth block	MxNx8	MxNx8
Reference texture block with extended taps	(3+8+4)x8x(M/8)x(N/8) x8	(3+4+4)x4x(M/4)x(N/4) x8
Compensated block	MxNx8	MxNx8

M is the block width and N is the block height in horizontal and vertical direction respectively; the bits of one pixel of texture or depth is 8 bits.

Complexity Assessment – Luma only

Number of operations						
	Proposal (Average)		Anchor		Ratio (to anchor)	
PU_size	Add/Sub/ Abs/Comp	Cons_Mul	Add/Sub/ Abs/Comp	Cons_Mul	Add/Sub/ Abs/Comp	Cons_Mul
64x64	34880	24576	32768	24576	106.45%	100.00%
64x32	17440	12288	16384	12288	106.45%	100.00%
32x64	17440	12288	16384	12288	106.45%	100.00%
32x32	8720	6144	8192	6144	106.45%	100.00%
32x16	4360	3072	4096	3072	106.45%	100.00%
16x32	4360	3072	4096	3072	106.45%	100.00%
16x16	2180	1536	2048	1536	106.45%	100.00%
16x8	1090	768	1024	768	106.45%	100.00%
8x16	1090	768	1024	768	106.45%	100.00%
8x8	545	384	512	384	106.45%	100.00%
8x4	256	192	256	192	100.00%	100.00%
4x8	256	192	256	192	100.00%	100.00%
Avg.					105.37%	100.00%

Complexity Assessment – Luma only

Data storage requirement (Bits)				
PU_size	Proposal (Average)	Anchor	Ratio (to anchor)	Difference bits (to anchor)
64x64	66604	65900	101.07%	704
64x32	33836	33132	102.12%	704
32x64	33836	33132	102.12%	704
32x32	17452	16748	104.20%	704
32x16	9260	8556	108.23%	704
16x32	9260	8556	108.23%	704
16x16	5164	4460	115.78%	704
16x8	3116	2412	129.19%	704
8x16	3116	2412	129.19%	704
8x8	2092	1388	150.72%	704
8x4	1100	876	125.57%	224
4x8	1324	876	151.14%	448
Average			118.96%	643

Complexity Assessment – Luma only

Data transfer rate (Bits/PU)			
PU_size	Proposal (Average)	Anchor	Ratio (to anchor)
64x64	136546	155660	87.72%
64x32	68279	86028	79.37%
32x64	68279	86028	79.37%
32x32	34146	43020	79.37%
32x16	17079	21516	79.38%
16x32	17079	21516	79.38%
16x16	8546	10764	79.39%
16x8	4279	5388	79.42%
8x16	4279	5388	79.42%
8x8	2146	2700	79.48%
8x4	1004	1356	74.04%
4x8	1228	1356	90.56%
Average			80.57%