|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  18th Meeting: Sapporo, JP, 30 June – 9 July 2014 | Document: JCTVC-R0165 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Non-SCCE1: Improved intra block copy coding with block vector derivation** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Yuwen He, Yan Ye, Xiaoyu Xiu  9710 Scranton Rd, #250 San Diego, CA 92121, USA | Tel: Email: | +1-858-210-4819/-4803/-4830 [yuwen.he@interdigital.com](mailto:yuwen.he@interdigital.com) [yan.ye@interdigital.com](mailto:yan.ye@interdigital.com) [xiaoyu.xiu@interdigital.com](mailto:xiaoyu.xiu@interdigital.com) |
| *Source:* | InterDigital Communications, Inc. | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This proposal proposes the block vector derivation method for improved intra block copy coding. The derived block vectors are used in intra block copy merge mode and normal intra block copy mode. Compared to SCCE1 full frame IntraBC anchors, the lossy coding of the proposed scheme reportedly achieves average {Y, U, V} BD rate gain of {-8.0%, -8.3%, -8.2%}, {-5.0%, -5.2%, -5.2%} and {-3.2%, -3.2%, -3.3%} for the the category (RGB/YUV, text & graphics with motion, 1080p and 720p) for AI, RA and LDB, respectively. And the lossless coding reportedly achieves total bit-rate saving of 4.8%, 3.0% and 2.9% for the category (RGB/YUV, text & graphics with motion, 1080p and text & graphics with motion, 720p) for AI, RA and LDB, respectively.

# Introduction

Various intra block copy (IntraBC) related technologies, including IntraBC merge mode, are investigated in SCCE1 [1]. This proposal proposes a new block vector (BV) derivation method. The derived BVs are used in two ways: (1) used in IntraBC merge mode as additional merge candidates to improve the merge mode performance; (2) used in normal IntraBC mode, where one additional flag is signaled to indicate whether the normal BV or the derived BV is used for IntraBC.

# The proposed block vector derivation

The proposed derived BV is obtained as follows for an IntraBC coded CU.

For a coding unit coded with IntraBC, a normal BV, referred to as BV0, is signaled. Denote the reference block pointed to by BV0 as block B1 (as shown in Figure 1). If the top-left sub-block of block B1 is also coded using IntraBC, then block B1 also has a BV of its own, which is referred to as BV1. In this case, the derived BV, referred to BVd, is calculated as follows:

BVd = BV0 + BV1 (1)



Figure . BV derivation if the reference block is IntraBC coded block

If the top-left sub-block of the reference block B1 is inter coded, then B1 has a motion vector of its own, which is referred to MV(B1). If B1 is coded with bi-prediction, then MV(B1) is selected as the motion vector associated with the higher quality reference picture (smaller QP). MV(B1) is in quarter pixel unit. In this case, the derived MV, MVd, in integer pixel unit is calculated as:

MVd = BV0 + ((MV(B1)+2) >>2) (2)

Note that in this case, the derived vector is a motion vector rather than a block vector, and is used for normal motion compensated prediction, and not for IntraBC prediction. Also note that the derived MV has integer pixel precision.



Figure . MV derivation if the reference block is inter coded block

If the referenced block B1 is coded using regular intra mode, then the proposed BV derivation method is not applicable, and the derived BV is not available.

## IntraBC merge mode using derived BV

The first way to use the derived BV is to add it as merge candidates for IntraBC. The maximum number of merge candidates for the proposed IntraBC merge mode is five. The merge candidate list is constructed with valid and unique BVs from the following possible candidates in order.

1. Last coded BV used for BV predictive coding;
2. Five spatial BVs from neighboring blocks (C0 to C4) shown in Figure 3;
3. For each BV in the list obtained from (1) and (2), if the corresponding reference block is IntraBC mode and the corresponding derived BV is unique and valid for the current PU, then add the derived BV to the merge candidate list. Derived MV in equation (2) is not used.



Figure . Spatial neighboring blocks for IntraBC merge candidate list construction

## IntraBC mode with derived block vector or derived motion vector

In normal IntraBC mode, its BV will be signaled for each prediction unit within coding unit. In the second way to use the derived BV, the normal IntraBC mode is extended by adding a flag. This flag indicates if the signaled block vector or the derived block vector/motion vector will be used in IntraBC prediction. If the flag is 0, then BV derivation process is not applied, and the signaled block vector is used directly for IntraBC prediction. If the flag is 1, then the derived BV or MV is calculated using Equation (1) or Equation (2) based on the signaled block vector, and the derived BV or MV (with integer pixel precision) is used for intra block copy prediction or motion compensated prediction.

# Simulation results

The compression performance of the proposed method is compared with SCCE1 anchors with the full frame IntraBC configuration, using the SCCE1 test conditions [1]. We did three tests as listed in Table 1. Table 2 and Table 3 give the average BD rate reduction for test C for lossless and lossy coding, respectively. Test A and B results are provided in the accompanying spreadsheets for further details.

Table . The settings of tests

|  |  |  |
| --- | --- | --- |
| **Test** | **IntraBC merge with BV derivation** | **Normal IntraBC with BV/MV derivation** |
| A | Enabled | Disabled |
| B | Disabled | Enabled |
| C | Enabled | Enabled |

As shown in Table 2, compared with SCCE1 anchors, for lossless coding, test C achieves total bit-rate saving of 7.7%, 5.5% and 5.6% for the category (RGB, text & graphics with motion, 1080p) for AI, RA and LDB, respectively. As shown in Table 3, for lossy coding, test C achieves average {Y, U, V} BD rate gain of {-9.6%, -9.7%, -9.8%}, {-6.1%, -6.0%, -6.1%} and {-4.9%, -4.7%, -4.8%} for the category (RGB, text & graphics with motion, 1080p) for AI, RA and LDB, respectively.

Table 2. Average BD rate reduction for test 3 lossless coding compared with SCCE1 full frame IntraBC anchors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **All Intra** | | | |
|  | Bit-rate saving (Total) | Bit-rate saving (Average) | Bit-rate saving (Min) | Bit-rate saving (Max) |
|  |
| RGB, text & graphics with motion, 1080p | 7.7% | 7.7% | 6.8% | 9.3% |
| RGB, text & graphics with motion,720p | 1.8% | 2.4% | 1.3% | 5.0% |
| RGB, mixed content, 1440p | 3.0% | 2.7% | 1.2% | 4.2% |
| RGB, mixed content, 1080p | 2.7% | 2.7% | 2.7% | 2.7% |
| RGB, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion, 1080p | 7.5% | 7.4% | 5.8% | 9.4% |
| YUV, text & graphics with motion,720p | 2.0% | 2.9% | 1.3% | 6.3% |
| YUV, mixed content, 1440p | 3.0% | 2.8% | 1.4% | 4.2% |
| YUV, mixed content, 1080p | 2.4% | 2.4% | 2.4% | 2.4% |
| YUV, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 128% | | | |
| Dec Time[%] | 96% | | | |
|  |  |  |  |  |
|  | **Random Access** | | | |
|  | Bit-rate saving (Total) | Bit-rate saving (Average) | Bit-rate saving (Min) | Bit-rate saving (Max) |
|  |
| RGB, text & graphics with motion, 1080p | 5.5% | 5.6% | 3.5% | 7.5% |
| RGB, text & graphics with motion,720p | 0.5% | 1.4% | 0.3% | 3.9% |
| RGB, mixed content, 1440p | 0.6% | 0.6% | 0.1% | 1.1% |
| RGB, mixed content, 1080p | 0.5% | 0.5% | 0.5% | 0.5% |
| RGB, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion, 1080p | 5.5% | 5.5% | 3.1% | 7.5% |
| YUV, text & graphics with motion,720p | 0.6% | 1.7% | 0.4% | 4.9% |
| YUV, mixed content, 1440p | 0.6% | 0.6% | 0.1% | 1.1% |
| YUV, mixed content, 1080p | 0.5% | 0.5% | 0.5% | 0.5% |
| YUV, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 104% | | | |
| Dec Time[%] | 100% | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  | **Low Delay B** | | | |
|  | Bit-rate saving (Total) | Bit-rate saving (Average) | Bit-rate saving (Min) | Bit-rate saving (Max) |
|  |
| RGB, text & graphics with motion, 1080p | 5.6% | 4.9% | 2.8% | 6.0% |
| RGB, text & graphics with motion,720p | 0.3% | 0.6% | 0.1% | 1.4% |
| RGB, mixed content, 1440p | 0.3% | 0.4% | 0.0% | 0.7% |
| RGB, mixed content, 1080p | 0.2% | 0.2% | 0.2% | 0.2% |
| RGB, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion, 1080p | 5.4% | 4.8% | 2.6% | 5.9% |
| YUV, text & graphics with motion,720p | 0.3% | 0.7% | 0.1% | 1.8% |
| YUV, mixed content, 1440p | 0.3% | 0.3% | 0.0% | 0.7% |
| YUV, mixed content, 1080p | 0.2% | 0.2% | 0.2% | 0.2% |
| YUV, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 103% | | | |
| Dec Time[%] | 100% | | | |

Table 3. Average BD rate reduction for test 3 lossy coding compared with SCCE1 full frame IntraBC anchors

|  |  |  |  |
| --- | --- | --- | --- |
|  | **All Intra** | | |
|  | G/Y | B/U | R/V |
| RGB, text & graphics with motion, 1080p | -9.6% | -9.7% | -9.8% |
| RGB, text & graphics with motion,720p | -6.4% | -6.7% | -6.6% |
| RGB, mixed content, 1440p | -4.1% | -4.0% | -4.0% |
| RGB, mixed content, 1080p | -4.2% | -4.1% | -4.1% |
| RGB, Animation, 720p | -0.2% | -0.2% | -0.2% |
| RGB, camera captured, 1080p | -0.1% | -0.1% | -0.1% |
| YUV, text & graphics with motion, 1080p | -9.5% | -9.8% | -9.7% |
| YUV, text & graphics with motion,720p | -6.6% | -6.9% | -6.9% |
| YUV, mixed content, 1440p | -4.1% | -4.3% | -4.3% |
| YUV, mixed content, 1080p | -4.4% | -4.5% | -4.5% |
| YUV, Animation, 720p | -0.3% | -0.4% | -0.4% |
| YUV, camera captured, 1080p | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 128% | | |
| Dec Time[%] | 90% | | |
|  |  |  |  |
|  | **Random Access** | | |
|  | G/Y | B/U | R/V |
| RGB, text & graphics with motion, 1080p | -6.1% | -6.0% | -6.1% |
| RGB, text & graphics with motion,720p | -4.0% | -4.1% | -4.2% |
| RGB, mixed content, 1440p | -2.5% | -2.5% | -2.5% |
| RGB, mixed content, 1080p | -2.7% | -2.9% | -2.8% |
| RGB, Animation, 720p | -0.2% | -0.2% | -0.2% |
| RGB, camera captured, 1080p | -0.1% | -0.1% | 0.0% |
| YUV, text & graphics with motion, 1080p | -5.9% | -6.2% | -6.2% |
| YUV, text & graphics with motion,720p | -4.1% | -4.3% | -4.2% |
| YUV, mixed content, 1440p | -2.5% | -2.9% | -3.0% |
| YUV, mixed content, 1080p | -3.2% | -3.3% | -3.1% |
| YUV, Animation, 720p | -0.2% | -0.4% | -0.2% |
| YUV, camera captured, 1080p | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 104% | | |
| Dec Time[%] | 99% | | |
|  |  |  |  |
|  | **Low delay B** | | |
|  | G/Y | B/U | R/V |
| RGB, text & graphics with motion, 1080p | -4.9% | -4.7% | -4.8% |
| RGB, text & graphics with motion,720p | -1.8% | -1.7% | -1.6% |
| RGB, mixed content, 1440p | -1.5% | -1.9% | -1.6% |
| RGB, mixed content, 1080p | -1.8% | -1.2% | -1.8% |
| RGB, Animation, 720p | 0.0% | -0.1% | 0.1% |
| RGB, camera captured, 1080p | -0.1% | 0.0% | 0.0% |
| YUV, text & graphics with motion, 1080p | -4.7% | -4.9% | -4.8% |
| YUV, text & graphics with motion,720p | -1.6% | -1.3% | -1.8% |
| YUV, mixed content, 1440p | -1.7% | -1.9% | -2.2% |
| YUV, mixed content, 1080p | -1.2% | -2.1% | -1.6% |
| YUV, Animation, 720p | -0.2% | -0.5% | -0.4% |
| YUV, camera captured, 1080p | 0.0% | 0.0% | -0.1% |
| Enc Time[%] | 101% | | |
| Dec Time[%] | 100% | | |

# Conclusions

In this proposal, the derived block vector method is proposed. The proposed derived BV is used to improve IntraBC merge mode and normal IntraBC mode. The simulations results compared to SCCE1 anchors show that the IntraBC coding can be greatly improved for both lossless and lossy coding with the proposed method.

# Syntax changes

#### The changes for IntraBC merge mode and additional flag signalling for normal IntraBC mode based on the draft specification P1005-v4 [2] of HEVC range extension are highlighted.

#### 7.3.8.5 Coding unit syntax

|  |  |
| --- | --- |
| coding\_unit( x0, y0, log2CbSize ) { | Descriptor |
| if( transquant\_bypass\_enabled\_flag ) |  |
| **cu\_transquant\_bypass\_flag** | ae(v) |
| if( slice\_type != I ) |  |
| **cu\_skip\_flag**[ x0 ][ y0 ] | ae(v) |
| nCbS = ( 1  <<  log2CbSize ) |  |
| if( cu\_skip\_flag[ x0 ][ y0 ] ) |  |
| prediction\_unit( x0, y0, nCbS, nCbS ) |  |
| else { |  |
| if( intra\_block\_copy\_enabled\_flag ) { |  |
| **intra\_bc\_flag**[ x0 ][ y0 ] | ae(v) |
| if(**intra\_bc\_flag**[ x0 ][ y0 ]) |  |
| **intra\_bc\_merge\_flag**[ x0 ][ y0 ] | ae(v) |
| } |  |
| if( slice\_type != I && !intra\_bc\_flag[ x0 ][ y0 ] ) |  |
| **pred\_mode\_flag** | ae(v) |
| if ( CuPredMode[ x0 ][ y0 ] != MODE\_INTRA | | (intra\_bc\_flag[ x0 ][ y0 ] &&  !**intra\_bc\_merge\_flag**[ x0 ][ y0 ] ) | | log2CbSize = = MinCbLog2SizeY ) |  |
| **part\_mode** | ae(v) |
| if( CuPredMode[ x0 ][ y0 ] = = MODE\_INTRA ) { |  |
| if( PartMode = = PART\_2Nx2N && pcm\_enabled\_flag &&   !intra\_bc\_flag[ x0 ][ y0 ] &&   log2CbSize >= Log2MinIpcmCbSizeY &&  log2CbSize <= Log2MaxIpcmCbSizeY ) |  |
| **pcm\_flag**[ x0 ][ y0 ] | ae(v) |
| if( pcm\_flag[ x0 ][ y0 ] ) { |  |
| while( !byte\_aligned( ) ) |  |
| **pcm\_alignment\_zero\_bit** | f(1) |
| pcm\_sample( x0, y0, log2CbSize ) |  |
| } else if( intra\_bc\_flag[ x0 ][ y0 ] ) { |  |
| if( intra\_bc\_merge\_flag[ x0 ][ y0 ] ) |  |
| **intra\_bc\_merge\_index**[ x0 ][ y0 ] | ae(v) |
| if( !intra\_bc\_merge\_flag[ x0 ][ y0 ] ) { |  |
| mvd\_coding( x0, y0, 2) |  |
| if( PartMode = = PART\_2NxN ) |  |
| mvd\_coding( x0, y0 + ( nCbS / 2 ), 2) |  |
| else if( PartMode = = PART\_Nx2N ) |  |
| mvd\_coding( x0 + ( nCbS / 2 ), y0, 2) |  |
| else if( PartMode = = PART\_NxN ) { |  |
| mvd\_coding( x0 + ( nCbS / 2 ), y0, 2) |  |
| mvd\_coding( x0, y0 + ( nCbS / 2 ), 2) |  |
| mvd\_coding( x0 + ( nCbS / 2 ), y0 + ( nCbS / 2 ), 2) |  |
| } |  |
| if( PartMode = = PART\_2Nx2N ) |  |
| **intra\_bc\_bv\_derivation\_flag**[ x0 ][ y0 ] | ae(v) |
| } |  |
| } else { |  |
| pbOffset = ( PartMode = = PART\_NxN ) ? ( nCbS / 2 ) : nCbS |  |
| for( j = 0; j < nCbS; j = j + pbOffset ) |  |
| for( i = 0; i < nCbS; i = i + pbOffset ) |  |
| **prev\_intra\_luma\_pred\_flag**[ x0 + i ][ y0 + j ] | ae(v) |
| for( j = 0; j < nCbS; j = j + pbOffset ) |  |
| for( i = 0; i < nCbS; i = i + pbOffset ) |  |
| if( prev\_intra\_luma\_pred\_flag[ x0 + i ][ y0 + j ] ) |  |
| **mpm\_idx**[ x0 + i ][ y0 + j ] | ae(v) |
| else |  |
| **rem\_intra\_luma\_pred\_mode**[ x0 + i ][ y0 + j ] | ae(v) |
| if( ChromaArrayType = = 3 ) |  |
| for( j = 0; j < nCbS; j = j + pbOffset ) |  |
| for( i = 0; i < nCbS; i = i + pbOffset ) |  |
| **intra\_chroma\_pred\_mode**[ x0 + i ][ y0 + j ] | ae(v) |
| else if( ChromaArrayType != 0 ) |  |
| **intra\_chroma\_pred\_mode**[ x0 ][ y0 ] | ae(v) |
| } |  |
| } else { |  |
| if( PartMode = = PART\_2Nx2N ) |  |
| prediction\_unit( x0, y0, nCbS, nCbS ) |  |
| else if( PartMode = = PART\_2NxN ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS / 2 ) |  |
| prediction\_unit( x0, y0 + ( nCbS / 2 ), nCbS, nCbS / 2 ) |  |
| } else if( PartMode = = PART\_Nx2N ) { |  |
| prediction\_unit( x0, y0, nCbS / 2, nCbS ) |  |
| prediction\_unit( x0 + ( nCbS / 2 ), y0, nCbS / 2, nCbS ) |  |
| } else if( PartMode = = PART\_2NxnU ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS / 4 ) |  |
| prediction\_unit( x0, y0 + ( nCbS / 4 ), nCbS, nCbS \* 3 / 4 ) |  |
| } else if( PartMode = = PART\_2NxnD ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS \* 3 / 4 ) |  |
| prediction\_unit( x0, y0 + ( nCbS \* 3 / 4 ), nCbS, nCbS / 4 ) |  |
| } else if( PartMode = = PART\_nLx2N ) { |  |
| prediction\_unit( x0, y0, nCbS / 4, nCbS ) |  |
| prediction\_unit( x0 + ( nCbS / 4 ), y0, nCbS \* 3 / 4, nCbS ) |  |
| } else if( PartMode = = PART\_nRx2N ) { |  |
| prediction\_unit( x0, y0, nCbS \* 3 / 4, nCbS ) |  |
| prediction\_unit( x0 + ( nCbS \* 3 / 4 ), y0, nCbS / 4, nCbS ) |  |
| } else { /\* PART\_NxN \*/ |  |
| prediction\_unit( x0, y0, nCbS / 2, nCbS / 2 ) |  |
| prediction\_unit( x0 + ( nCbS / 2 ), y0, nCbS / 2, nCbS / 2 ) |  |
| prediction\_unit( x0, y0 + ( nCbS / 2 ), nCbS / 2, nCbS / 2 ) |  |
| prediction\_unit( x0 + ( nCbS / 2 ), y0 + ( nCbS / 2 ), nCbS / 2, nCbS / 2 ) |  |
| } |  |
| } |  |
| if( !pcm\_flag[ x0 ][ y0 ] ) { |  |
| if( CuPredMode[ x0 ][ y0 ] != MODE\_INTRA &&   !( PartMode = = PART\_2Nx2N && merge\_flag[ x0 ][ y0 ] ) | |   ( CuPredMode[ x0 ][ y0 ] = = MODE\_INTRA && intra\_bc\_flag[ x0 ][ y0 ] ) ) |  |
| **rqt\_root\_cbf** | ae(v) |
| if( rqt\_root\_cbf ) { |  |
| MaxTrafoDepth = ( CuPredMode[ x0 ][ y0 ] = = MODE\_INTRA ?   ( max\_transform\_hierarchy\_depth\_intra + IntraSplitFlag ) :   max\_transform\_hierarchy\_depth\_inter ) |  |
| transform\_tree( x0, y0, x0, y0, log2CbSize, 0, 0 ) |  |
| } |  |
| } |  |
| } |  |
| } |  |

**intra\_bc\_merge\_flag**[ x0 ][ y0 ] equal to 1 specifies that the current coding unit is coded in merge mode, and the block vector is derived from the merge candidates. intra\_bc\_merge\_flag[ x0 ][ y0 ] equal to 0 specifies that the coding unit is not coded in merge mode, and block vector of current coding unit is coded explicitly. When not present, the value of intra\_bc\_merge\_flag is inferred to be equal to 0. The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture.

**intra\_bc\_merge\_index**[ x0 ][ y0 ] specifies the index among merge candidates that the block vector of current coding unit is the same as. intra\_bc\_merge\_index[ x0 ][ y0 ] shall be in the range of 0 to maximal number of intra block copying merge candidates minus 1. The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture.

**intra\_bc\_bv\_derivation\_flag**[ x0 ][ y0 ] equal to 1 specifies that the derived BV or MV is used for current PU prediction.

# Patent rights declaration(s)

**InterDigital Communications, Inc. may have IPR relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

# References

1. J. Sole, S. Liu, “HEVC Screen Content Coding Core Experiment 1 (SCCE1): Intra Block Copying Extensions”, JCTVC-Q1121, Apr. 2014.
2. D. Flynn, M. Naccari, K.Sharman, C. Rosewarne, J. Sole, G. J. Sullivan, T. Suzuki, “HEVC Range Extension Draft 6”, JCTVC-P1005, Jan. 2014, San Jose.