Video compression
Beyond H.264, HEVC

O. Le Meur
olemeur@irisa.fr

Univ. of Rennes 1
http://www.irisa.fr/temics/staff/lemeur/

November 7, 2011

Introduction

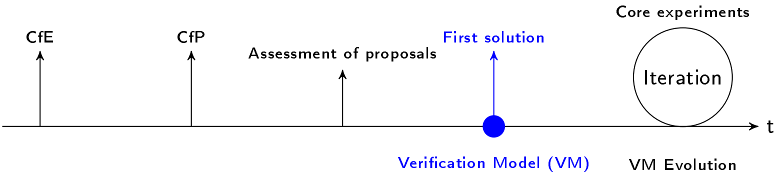
What is the goal of a compression standard?

Definition (Standard)

A format that has been approved by a recognized standards organization or is accepted as a de facto standard by the industry.

For the compression standards, there are two main organizations: ITU-T-VCEG and ISO MPEG. A video compression standard only specifies bitstream syntax and decoding process.

The goal is to create the best video compression standards for targeted applications.

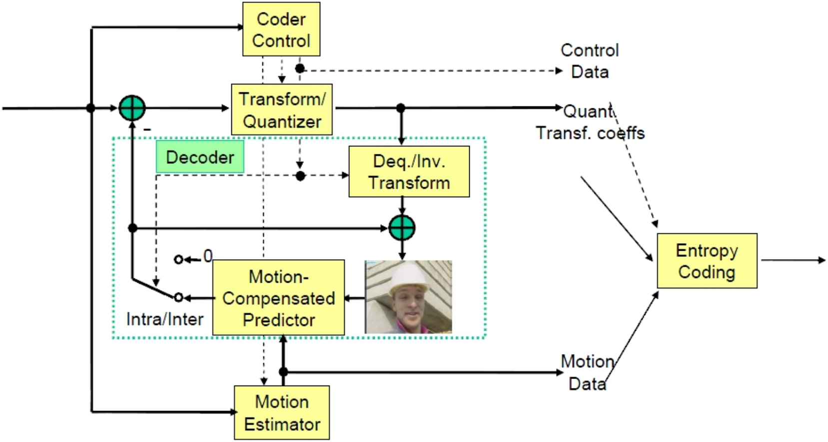


CfE: Call for Evidence; CfP: Call for Proposal.

For example: H264, CfP (1998), standard 2003.

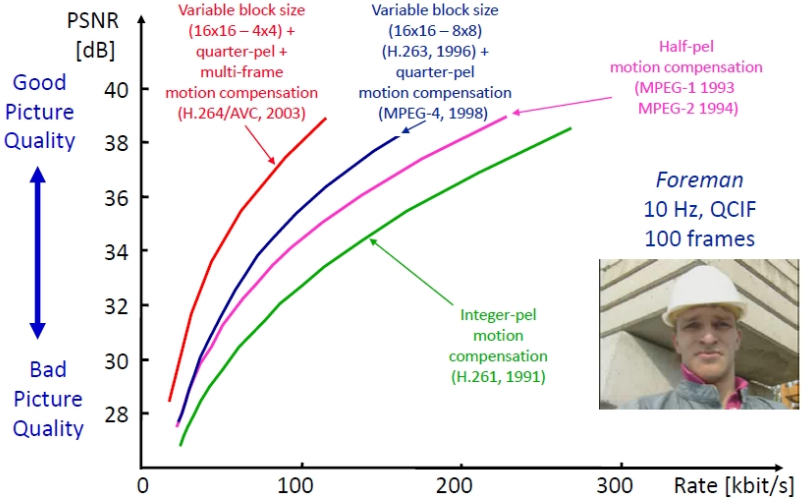
A common framework for the different video standards (H.261, MPEG-1, MPEG-2, H.263, MPEG-4, H.264/AVC):

Motion-compensated hybrid coding[1]



[1] Extracted from B. Girod's courses (EE398 Image and Video Compression).

Performance



[2]

[2] Extracted from G. Sullivan's presentation at IEEE Intl. Conf. on Consumer Electronics 2011

H.264 or MPEG-4 AVC

ITU-T H.264 / ISO/IEC 14496-10 MPEG-4 AVC
Basic Milestones

- May 2003: first version of standard
- Mid 2004: fidely range extensions (High Profile)
- Mid 2006: extended-gamut color spaces
- Mid 2006: professional profiles
- Fall 2007: scalable video coding (SVC) extension
- Fall 2008: multi-view video coding (MVC) extension

Motivations

They are always the same !!!

- due to the higher demand for higher resolution videos, HEVC aims at achieving a higher video quality than what H.264/MPEG-4 AVC (Advanced Video Coding) standard presents.
- a new generation of video compression technology with high compression capability reduce by 50% the data rate needed for high quality video coding, compared AVC standard
- be capable of trading off complexity and compression capability

HEVC call

HEVC call

The timeline of the Call for Proposals is as follows:

- 2010/01/22: Final Call for Proposals
- 2010/01/22: Formal registration period opens.
- 2010/02/15: Formal registration period ends
- 2010/02/22: Coded test material shall be available at the test site . By this date, the payment of the testing fee is expected to be finalized.
- 2010/03/02: Subjective assessment starts
- 2010/04/12: Registration of documents describing the proposals
- 2010/04/13: Submission of documents
- 2010/04/15: Cross-checking of bitstreams and binary decoders (participation mandatory for proponents)
- 2010/04/16: Subjective test results available within standardization body
- 2010/04/16-23: Evaluation of proposals at standardization meeting

Anticipated tentative timeline after CfP:

- Test model selection process begins 2010/04
- Test model selection by 2010/10
- Final standard approval by 2012/07

HEVC call

Test Classes and Bit Rates (18 sequences, YUV, 420, 8 bits per sample):

Class	Bit Rate 1	Bit Rate 2	Bit Rate 3	Bit Rate 4	Bit Rate 5
A: 2560x1600p30	2.5 Mbit/s	3.5 Mbit/s	5 Mbit/s	8 Mbit/s	14 Mbit/s
B1: 1080p24	1 Mbit/s	1.6 Mbit/s	2.5 Mbit/s	4 Mbit/s	6 Mbit/s
B2: 1080p50-60	2 Mbit/s	3 Mbit/s	4.5 Mbit/s	7 Mbit/s	10 Mbit/s
C: WVGAp30-60	384 kbit/s	512 kbit/s	768 kbit/s	1.2 Mbit/s	2 Mbit/s
D: WQVGAp30-60	256 kbit/s	384 kbit/s	512 kbit/s	850 kbit/s	1.5 Mbit/s
E: 720p60	256 kbit/s	384 kbit/s	512 kbit/s	850 kbit/s	1.5 Mbit/s

Constraint cases are defined as follows:

- Constraint set 1: structural delay of processing units not larger than 8-picture groups of pictures (GOPs) (e.g., dyadic hierarchical B usage with 4 levels), and random access intervals of 1.1 seconds or less.
- Constraint set 2: no picture reordering between decoder processing and output, with bit rate fluctuation characteristics and any frame-level multi-pass encoding techniques to be described with the proposal. (A metric to measure bit rate fluctuation is implemented in the Excel file to be submitted for each proposal.)

HEVC call

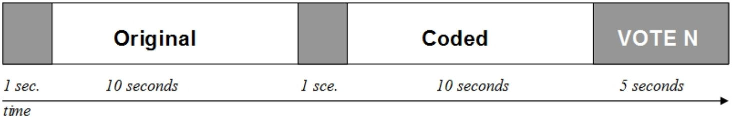
Submissions to the call shall obey the following additional constraints:

- No use of pre-processing.
- Only use post-processing if it is part of the decoding process, i.e. any processing that is applied to a picture prior to its use as a reference for inter prediction of other pictures. Such processing can also be applied to non-reference pictures.
- Quantization settings should be kept static. When change of quantization is used it shall be described.
- Proponents are discouraged from optimizing encoding parameters using non-automatic means.
- The video coding test set shall not be used as the training set for training large entropy coding tables, VQ codebooks, etc.

HEVC call

- Submission requirements: All proponents needed to deliver, by the due date, a hard drive to the address of the Test Coordinator. For Classes B, C, D and E, the disk shall contain the bitstreams, YUV and AVI files, as well as a decoder executable used by the proponent to generate the YUV files from the bitstreams.

- The proposals submission material are evaluated by means of a formal subjective assessment process. The tests were conducted at FUB (Test Coordinator, Rome, Italy), EBU (Geneva, Switzerland) and EPFL (Lausanne, Switzerland).
  The anticipated test methods are:
  1. DSIS (Double Stimulus Impairment Scale)



  2. DSCQS (Double Stimulus Continuous Quality Scale)

HEVC call

Anchors have been generated by encoding the above sequences using an AVC encoder:

- useful reference points demonstrating the behaviour of well-understood configurations of current technology

Example: Alpha anchor (satisfies constraint set 1)

- Conformance with High Profile
- Hierarchical B pictures IbBbBbBbP (8) coding structure - each picture uses at most 4 reference pictures in each list for inter prediction
- Open GOP structuring with an Intra picture every 24, 32, 48 and 64 pictures for 24 fps, 30 fps, 50 and 60 fps sequences, respectively
- maxRefFrames = 4
- QP scaling: QP (I picture), QP+1 (P picture), QP+2 (first B layer), QP+3 (second B layer), QP+4 (third B layer)
- CABAC, 8x8 transforms enabled
- Flat quantization weighting matrices
- RD Optimization enabled
- RDOQ enabled (fast mode, NUM=1)
- Adaptive rounding disabled
- Weighted prediction enabled
- Fast motion estimation (range 128x128)

CfP's results

CfP's results

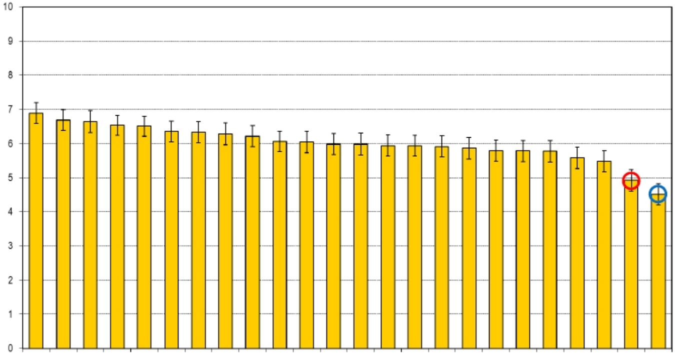
- 27 complete proposals submitted (some multi-organizational): France Telecom, NTT, NTT DOCOMO, Panasonic, Technicolor, HHI, Tadberg, Ericsson, Nokia, RIM, Qualcomm, Samsung, BBC...)
- Each proposal was a major package -lots of encoded video, extensive documentation, extensive performance metric submissions, sometimes software, etc.
- Extensive subjective testing (3 test labs, 4 200 video clips evaluated, 850 human subjects, 300 000 scores)
- Quality of proposal video was compared to AVC anchor encodings.

In a number of cases, comparable quality at half bit rate.

CfP's results

Overall average MOS results over all Classes for Random Access coding conditions

- 11 grade scale (0 represents the worst and 10 represents the best quality).
- Alpha anchor was tested twice and the two results are indicated by the two right-most bars



A significant quality gap can be observed between the AVC anchors and most proposals.

CfP's results

The best-performing proposals in a significant number of cases showed similar quality as the AVC anchors when encoded at roughly half of the anchor bit rate.

All proposals basically conceptually similar to AVC (and prior standards)

- Block-based
- Variable block sizes
- Block motion compensation
- Fractional-pelmotion vectors
- Spatial intra prediction
- Spatial transform of residual difference
- Integer-based transform designs
- Arithmetic or VLC-based entropy coding
- In-loop filtering to form final decoded picture

Lots of variations at the individual tool level

Details of HEVC

Main features

A new framework composed of three new concepts:

- Coding unit
- Prediction unit
- Transform unit

Goal: to be as flexible as possible and to adapt the compression-prediction to image peculiarities.

Coding Units

Introduction of larger block structures with flexible mechanisms of sub-partitioning.

Definition (Coding Unit)

Coding units (CUs) define a sub-partitioning of a picture into rectangular regions of equal or (typically) variable size. The coding unit replaces the macroblock structure as known from previous video coding standards, and contains one or several prediction unit(s) (PUs) and transform units (TUs).

→ The basic partition geometry of all these elements is encoded by a scheme similar to the well-known quad-tree segmentation structure.

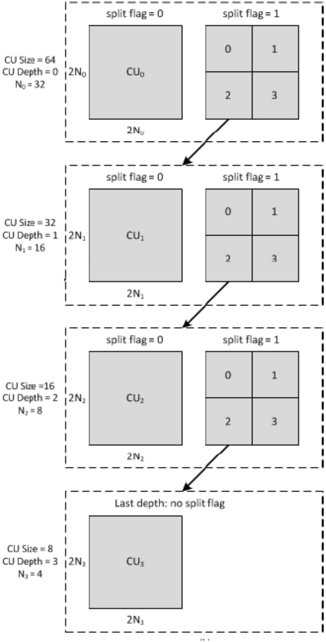
→ At the level of PU, either intra-picture or inter-picture prediction is selected.

Coding unit tree structure is limited from 8x8 to 64x64 for luma, that is to say, no splitting is allowed for CU3.

A CU is distinguished by 2 properties:

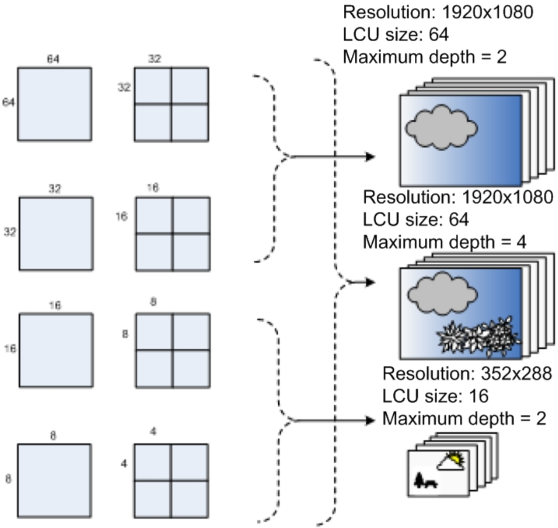
- its LCU (Largest CU) size
- the hierarchical depth in the LCU that the CU belongs to.

NB: if the LCU size = 16 and hierarchical depth = 2, then this is a similar coding structure to macro-block and sub-macro-block in H.264/AVC.



Coding Units

Example of LCU size and maximum depth combinations for various resolutions

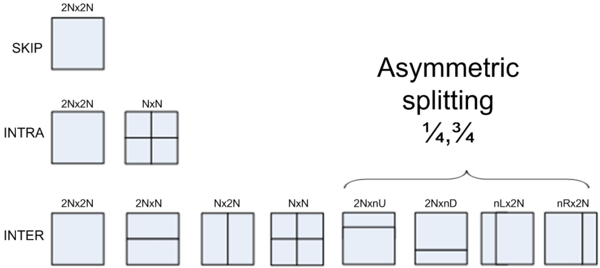


Prediction Unit

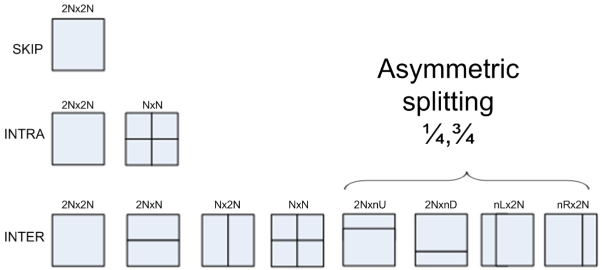
Definition (Prediction Unit)

PU is basically **the elementary unit for prediction**. PUs are defined after the last level of CU splitting i.e. at the very last level in the CU tree. Prediction type and PU splitting are two concepts that describe the prediction method.

- Largest allowed PU size is equal to the CU size
- Other allowed PU sizes depend on prediction type



Prediction Unit

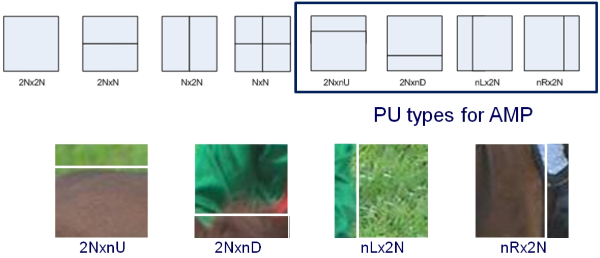


Example of 128x128 CU:

- Skip: PU = 128x128
- Intra: PU = 128x128 or 64x64
- Inter: PU = 128x128, 128x64, 64x128, 64x64, 128x32, 128x96, 32x128 or 96x128

Prediction Unit

Asymmetric splitting can be very useful as illustrated below:



- Computationally efficient compared to non-rectangular partitions
- Different object motions can be handled without further splitting (Asymmetric motion partition).

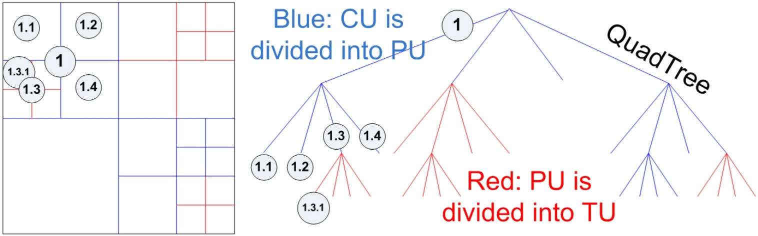
Transform Unit

Definition (Transform Unit)

The transform unit (TU) is **the unit for transform and quantization**.

May exceed size of PU, but not CU...

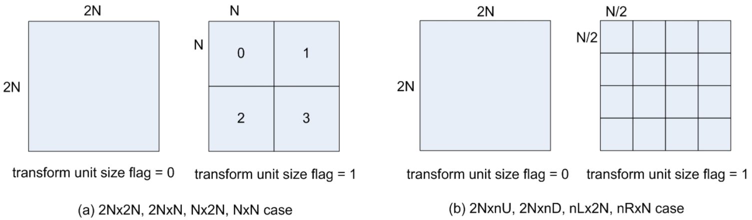
- A TU has a size from 4×4 to 32×32 (for luminance)
- A PU might contain more than one TU as illustrated below (They are arranged in a quadtree structure):



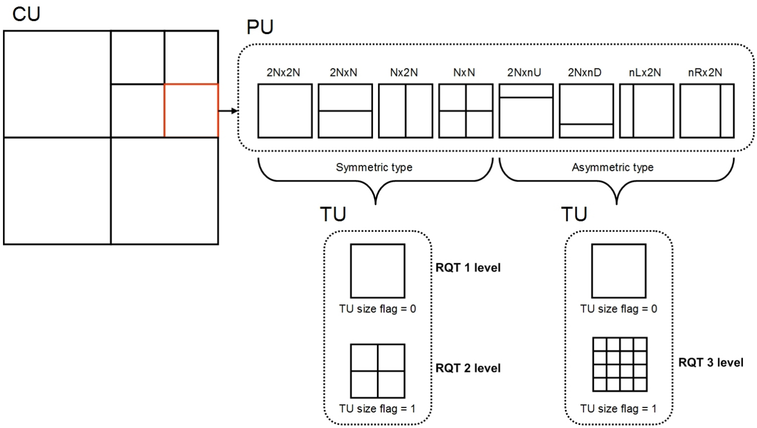
Transform Unit

Only two TU options are allowed, signalled by transform unit size flag:

- The size of the TU is the same as its PU if the flag is set to 0.
- Otherwise the size is set to either $N \times N$ (PU splitting is symmetric), $\frac{N}{2} \times \frac{N}{2}$ (PU splitting is asymmetric) or non-square (See NSQT).



Relationship of CU, PU and TU



Extracted from JCTVC-A124.

Residual Quadtree Transform = RQT

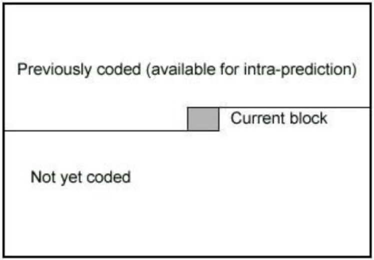
Intra Prediction

Intra Prediction

Definition (Goal of intra-prediction)

To exploit spatial correlation among pixels in order to reduce the amount of data to transmit.

A prediction mode defines a method for generating a signal from previously encoded data, i.e. either spatial or temporal, that minimizes the residual between prediction and original.



Intra Prediction

At the level of PU, intra-prediction is performed from samples already decoded in adjacent PUs. Different modes can be used:

- DC (average)
- one of up to 33 angular directions depending on the size of the corresponding PU:

PU size	Number of UDI modes
4	17
8	34
16	34
32	34
64	3

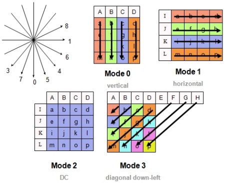
UDI = Unified Directional Intra

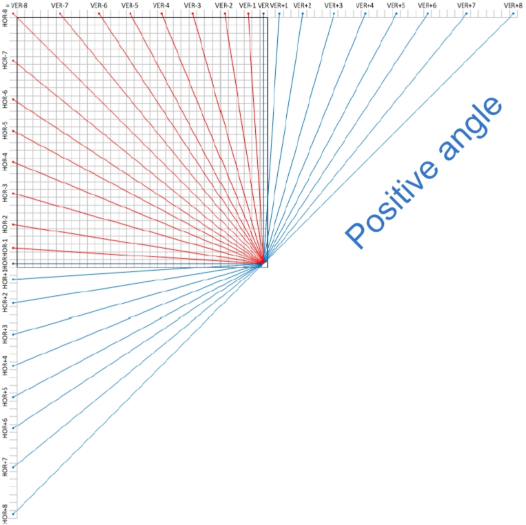
- Planar mode
- SDIP (Short Distance Intra Prediction)
- MDIS (Mode Dependent Intra Smoothing)

Angular Intra Prediction

The 33 possible intra predictions are illustrated below. The prediction directions in the intra prediction have the angles of $\pm$ [0, 2, 5, 9, 13, 17, 21, 26, 32]:

H264





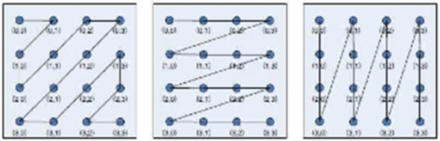
Angular Intra Prediction

- Two arrays of reference samples are used:
  - → the row of samples lying above the current PU to be predicted
  - → the column of samples lying to the left of the same PU
- One of the reference arrays is defined to be the main array and the other array the side array:
  - → vertical prediction: the reference row above the PU is called the main array and the reference column to the left of the same PU is called the side array.
  - → horizontal prediction: the reference column to the left of the PU is called the main array and the reference row above the PU is called the side array.
- Predicted pixels are computed by using the linear interpolation of the reference top or left samples.
- Only the main array samples are used for prediction when the intra prediction angle is positive
- When the intra prediction angle is negative, a per-sample test is performed to determine whether samples from the main or the side array should be used for prediction

Angular Intra Prediction

Mode Dependent Coefficient Scan (MDCS):

- 3 different scan patterns used to improve the residual coding:



0 : Zigzag scan   1 : Horizontal scan   2 : Vertical scan

	Mode Luma																																	
TU	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
32x32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16x16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8x8	1	2	0	0	1	1	0	2	2	0	0	1	1	0	0	2	2	0	0	0	1	1	1	1	0	0	0	0	2	2	2	2	0	0
4x4	1	2	0	0	1	1	0	2	2	0	0	1	1	0	0	2	2	0	0	0	1	1	1	1	0	0	0	0	2	2	2	2	0	0
2x2																																		

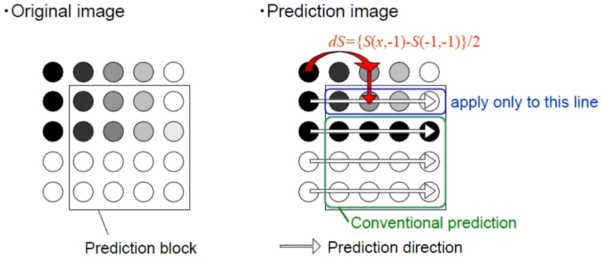
For luma component

Angular Intra Prediction

Improved intra vertical and horizontal prediction (JCTVC-F172):

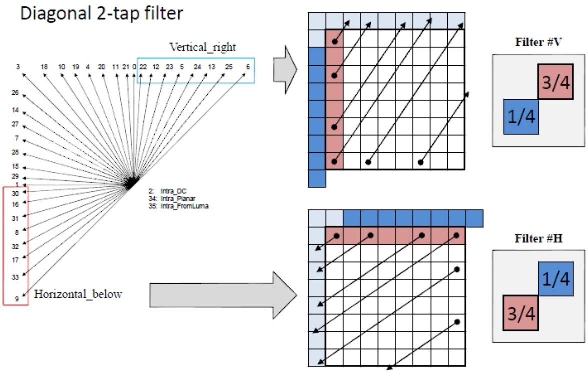
Horizontal prediction:

$$S'(x,y)=\begin{cases} S(-1,y)+(S(x,-1)-S(-1,-1))/2 & (y=0) \\ S(-1,y) & (y\geq 1) \end{cases}$$



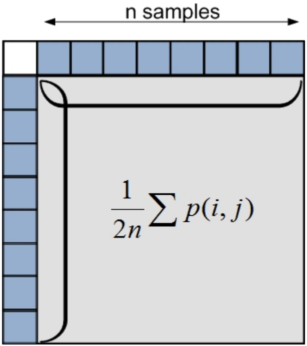
Angular Intra Prediction

To improve the prediction continuity in the boundary region (JCTVC-F358):



DC mode

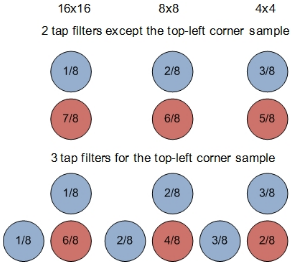
In DC mode, the mean values of samples from both top row and left column are used for the DC prediction.



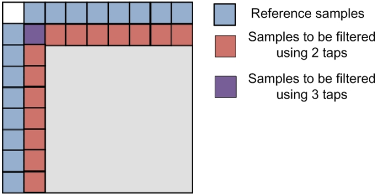
DC mode

DC Prediction Filtering:

- only top and left edges of DC prediction are filtered
- one of 3 filters is selected according to block size

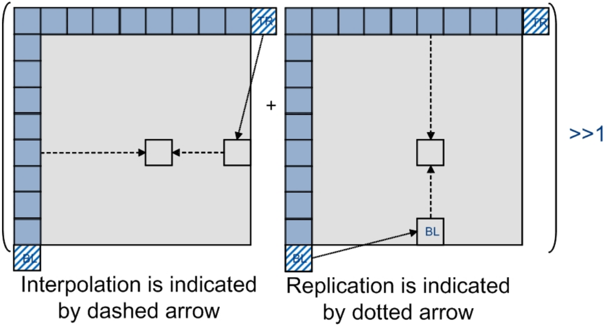


Example: 8x8 DC prediction



Planar Intra Prediction

Instead of performing a simple propagation (horizontal, vertical), the planar mode could be used to predict more efficiently textured areas.



Different solutions were evaluated, including bilinear interpolation (JCTVC-D326)...

SDIP (JCTVC-E278)

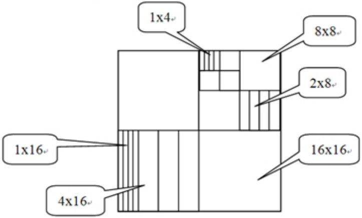
Definition (SDIP = Short Distance Intra Prediction Method)

In SDIP, one $N \times N$ square block is divided into several lines or non-square blocks with rectangle shape. In the block, pixels are predicted and reconstructed line by line or rectangle by rectangle.

- to better exploit spatial correlations
- to reduce the energy of the prediction residuals by reducing the distance of predicted pixel and its reference pixels

For 32×32 CU, only rectangular SDIP PU is used. For 16×16 and 8×8 CU, both the rectangular and line based PU are supported because there are more textures in these kind of CUs.

SDIP block partitions



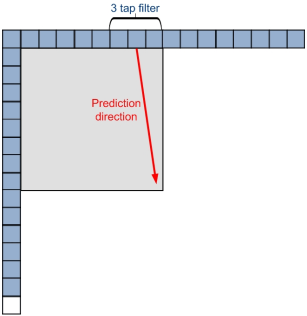
- 32×32 CU $\rightarrow 8 \times 32, 32 \times 8$
- 16×16 CU $\rightarrow 4 \times 16, 16 \times 4 \rightarrow 1 \times 16, 16 \times 1$
- 8×8 CU
  $\rightarrow 2 \times 8, 8 \times 2$
  $\rightarrow 1 \times 16, 16 \times 1$

Mode Dependent Intra Smoothing (JCTVC-D282)

Definition (Mode Dependent Intra Smoothing)

To reduce high frequencies in samples that are used for the prediction. Reference pixels, as illustrated below, are filtered.

- 3-tap filter
- Filtering decision is based on **prediction unit (PU) size** and **prediction direction**
  $\Rightarrow$ Post processing...
- Given an intra prediction mode and PU size:
  - → Most probable filter (LUT)
  - → Second most probable filter (LUT)
    Intra smoothing filter is selected between these two filters
- The encoder makes the smoothing decision based on filtered by testing the RD cost of intra prediction based on **filtered** or **unfiltered prediction** samples.



Mode Dependent Intra Smoothing

LUT most probable filter

Most probable filter table: 0 – no filter, 1 – filter 1, 2 – filter 2

TU \ Mode	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
2x2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4x4	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8x8	0	0	0	1	1	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16x16	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32x32	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
64x64	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
128x128	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Second most probable filter table: 0 – no filter, 1 – filter 1, 2 – filter 2

TU \ Mode	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
2x2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4x4	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8x8	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16x16	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
32x32	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64x64	2	2	2	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
128x128	2	2	2	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Combined with the no filtering case (filter 0), there are three filtering cases. For each prediction block, we select the suitable filter among the three cases (filter 0, filter 1, and filter 2).

Constrained intra prediction (JCTVC-E488)

Definition (CIP)

The aim of constraining the intra prediction process is to increase error resilience by avoiding the use of neighboring inter macroblock residual data and decoded samples for the prediction of intra macroblocks. This way the loss of data will not affect negatively intra predicted macroblocks, which results in an effective method to block the propagation of errors.

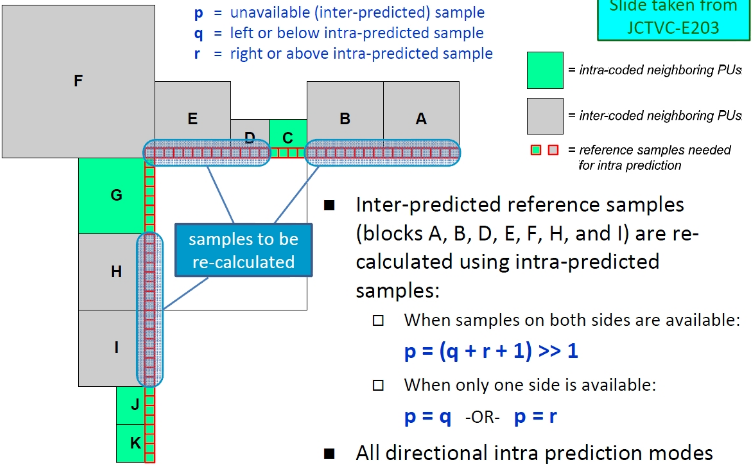
- avoids spatial noise propagations caused by spatial intra prediction with encoder-decoder mismatched reference pixels (packet loss...)
- Robust video transmission over unreliable networks

Intra prediction reference samples are marked as available or unavailable for prediction:

- the sample is outside the picture
- the sample is outside the slice
- the sample is not intra coded and CIP is enabled

If all reference samples are unavailable, we set all prediction pixels to 128. Otherwise, we use either sample repetition or lightweight sample interpolation

Constrained intra prediction



Slide taken from JCTVC-E203

- Inter-predicted reference samples (blocks A, B, D, E, F, H, and I) are re-calculated using intra-predicted samples:
  - When samples on both sides are available:

    p = (q + r + 1) >> 1
  - When only one side is available:

    p = q -OR- **p = r**
- All directional intra prediction modes remain available to be used.

Inter Prediction

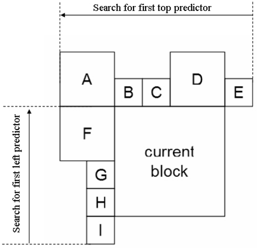
For an inter prediction both residual data in form of quantized transform coefficients and motion vector information that points to previously encoded/decoded pictures are transmitted.

- Prediction computation:
  - → 1/4 luma sample, 12-tap DCT-based interpolation filter (high efficiency configuration)
  - → 1/4 luma sample, 6-tap directional interpolation filter (low complexity configuration)
  - → 1/8 chroma sample, bilinear interpolation filter for (both HE and LC)
- Motion vector coding:
  - → advanced motion vector prediction
  - → CU merging + CU skip/direct

Motion Vector Coding

Three PMV candidates:

- a left predictor
- a top predictor
- a collocated predictor.



- For the left predictor: the first available motion vector is searched from bottom to top.
- For the top predictor: same as previous but from the right to the left.
- For the collocated predictor: it is derived from the collocated block.

A motion vector is considered **available** if the vector exists (for the same direction ($L0$ or $L1$) and uses the same reference index as the current block).

Skip mode

SKIP mode = no transmission of residual data

The skip mode uses the AMVP method to derive motion information.

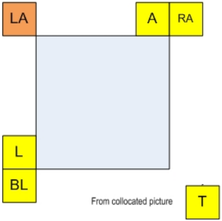
- the motion vector(s) is(are) equal to the MVP(s) and no residual is transmitted.
- Motion compensation is performed with a motion vector precision up to quarter-sample precision.

Merge mode (JCTVC-F297)

If the skip mode uses the MERGE method derive motion information, a merge index is signaled. The motion information is fully derived from the neighboring block indexed by the merge index and no residual is transmitted either.

The checking procedure of Merge mode is as follows:

1. Check L and BL
2. Check A and RA
3. If either the candidate at L or BL is invalid, use the candidate at LA (if valid) to replace the invalid one (L has higher priority than BL to be replaced if both are invalid).
4. If LA is not used in step 3 and if either the candidate at A or RA is invalid, use the candidate at LA (if valid) to replace the invalid one (A has higher priority than RA to be replaced if both are invalid).



Transform

4×4 and 8×8 partitions

Same as H.264 for 4×4 and 8×8 partitions.

Three transforms depending on the type of residual data:

- A Hadamard transform for the 4×4 array of luma DC coefficients in Intra MB predicted in 16×16 mode;
- A Hadamard transform for the 2×2 array of chroma DC coefficients;
- A DCT-based transform for all other 4×4 blocks in the residual data.

16×16 and 32×32 partitions

Chen's fast DCT algorithm is used (Butterfly structure).

Non Square Quadtree Transform(NSQT)

When a transform block cross the boundary of motion block, high frequency coefficients are generated which will impose negative influence on coding performance.

- Implicit TU split method (JCTVC-E364):
  - → If PU is a square block, TU is a square block of the same size.
  - → If PU is a rectangular block, TU size is then set to the size of the largest possible square block that can fit into the PU.
- Non-square TU is used when partition type is asymmetric partition ($2N \times N$ and $N \times 2N$).

Considering PU partition type can reflect local region's texture property, non-square TU can also be used at symmetric motion partitions. The size of a transform unit is tied to PU partition type.

Transform

In general, a $n \times m$ block is transformed by using the following formula:

$$C_{n\times m} = T_m \times B_{n\times m} \times T_n^T$$

- $B_{n\times m}$ denotes a block with n pixels m rows;
- T_n and T_m are the matrices of size $n \times n$ and $m \times m$, respectively;
- $C_{n\times m}$ denotes the transformed $n \times m$ block.

$$C_{2M\times M/2} = T_{M/2} \times B_{2M\times M/2} \times T_{2M}^T \qquad C_{M/2\times 2M} = T_{2M} \times B_{M/2\times 2M} \times T_{M/2}^T$$

Non-square TU is used at 32x8, 8x32, 16x4 and 4x16 asymmetric partition at luma component, and is used at 16x4, 4x16, 8x2 and 2x8 at chroma component.

Loop filters

- Deblocking loop filter (same as AVC)
- Adaptive loop filter for high efficiency configuration

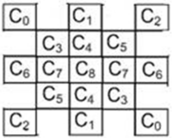
Adaptive Loop Filter

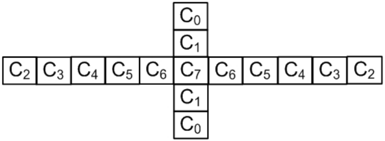
Each TU is processed by a spatial block transform and quantization of the resulting transform coefficients.

Definition (Adaptive Loop Filter (JCTVC-F303))

ALF is applied within the prediction loop prior to copying the frame into a reference decoded picture buffer to provide improved objective and subjective quality.

For each CU, one from the two shapes (Star 5×5, Cross 11×5) which provide the better coding efficiency (rate-distortion) is selected.

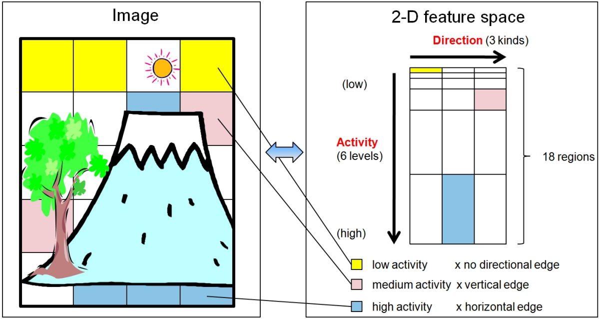




Adaptive Loop Filter (JCTVC-E323)

Image is divided into up to 18 regions:

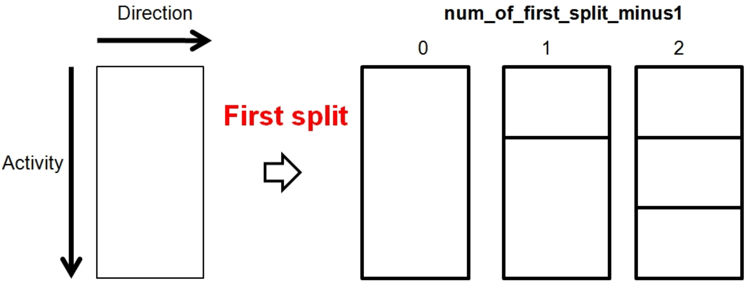
- on a 4 × 4 block accuracy
- depending on local activity (Laplacian, 6 levels) and Direction (up to 3 kinds)



Extracted from JCTVC-D116

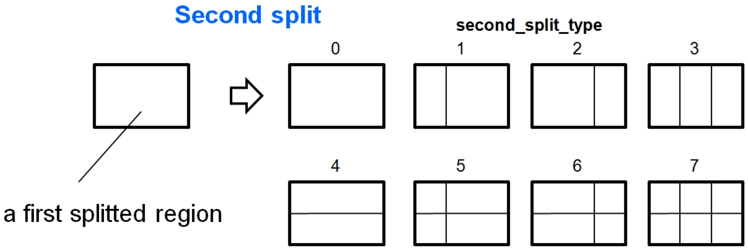
Adaptive Loop Filter (JCTVC-E323)

- First split:
  Image is divided using Activity into up to 3 regions



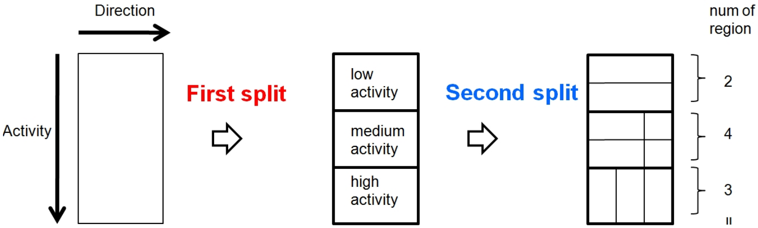
Adaptive Loop Filter (JCTVC-E323)

- Second split:
  Each first splitted region is further divided using Activity and Direction into up to 6 regions



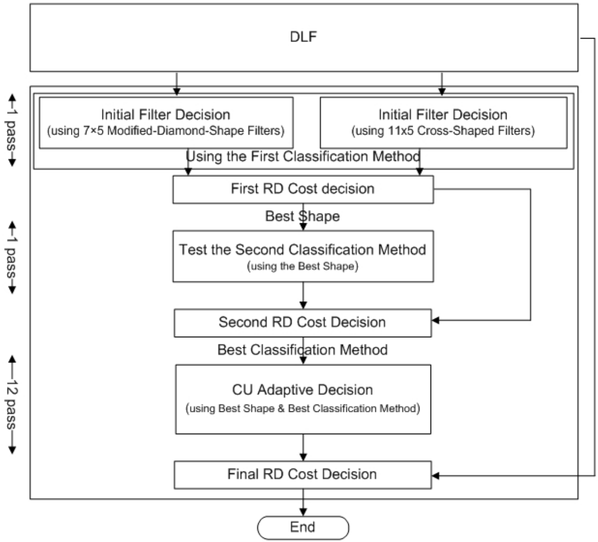
Adaptive Loop Filter (JCTVC-E323)

Hierarchical region division based on two-step division first split and second split.



Adaptive Loop Filter (JCTVC-E323)

Global decision:



Quantization

The quantizer, same as H.264, is applied on a CU.
In AVC, the quantization step size increases by approximately 12.25% with each increment of QP which lead to average around 19% and up to 44.2% bit increase.
For adaptive quantization algorithm like TM-5, there are some new proposals related to ΔQP coding (JCTVC-F024):

- Previous QP
- Left or above QP
- Median / Average
- PredMode-dependent...
- Temporal QP prediction with motion vector

The last proposition provides best result in coding efficiency, but concerns are raised if it works with real-world applications...

Entropy coding

Suggestion for further reading...

S. Daly. The visible differences predictor: An algorithm for the assessment of image fidelity. Digital Images and Human Vision, pp. 179-206, 1993, MIT Press.