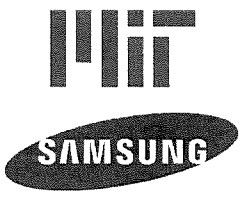
ECE 634: Digital Video Systems
Video coding standards: 3/2/17

Part 2

Professor Amy Reibman
MSEE 356
reibman@purdue.edu

http://engineering.purdue.edu/~reibman/ece634/index.html



Slides with ← are taken from here →

Design and Implementation of Next Generation Video Coding Systems (H.265/HEVC Tutorial)

Vivienne Sze (sze@mit.edu)
Madhukar Budagavi (m.budagavi@samsung.com)

ISCAS Tutorial 2014

Overview of the High Efficiency Video Coding (HEVC) Standard

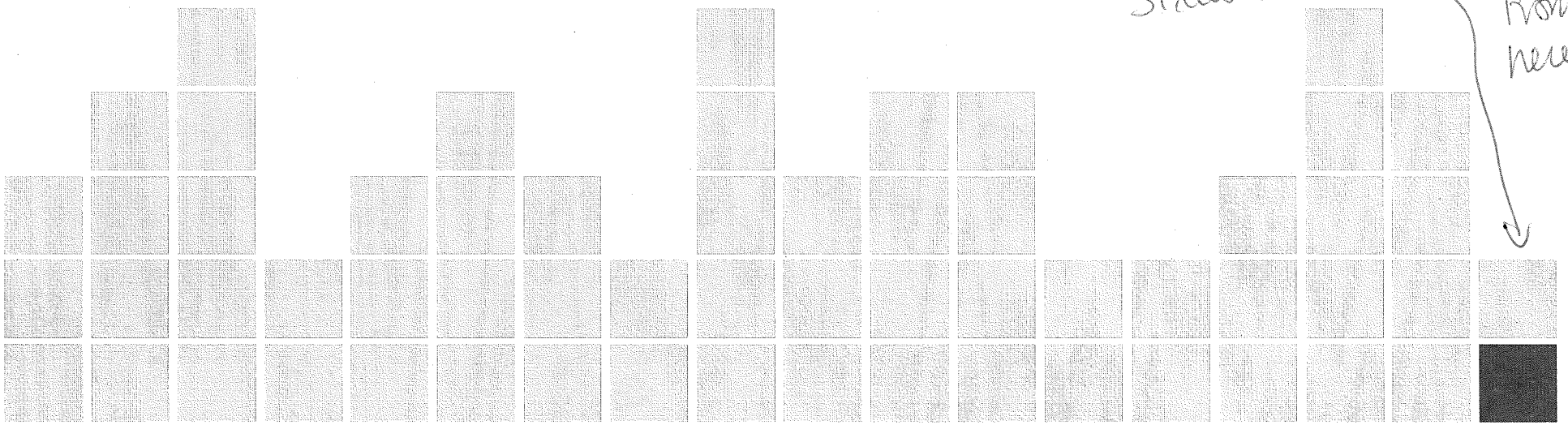
G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand

IEEE Trans. Circuits and Systems for Video Technology, vol. 22, no. 12, Dec., 2012

Gaewon Kim (Ph.D. course) and Prof. Changhoon Yim

Department of Internet and Multimedia Engineering, Konkuk University

Slides with are taken from here



Overview: Standards evolution

- Standardization Timeline; requirements
    - Bit-rates, spatial resolution, etc
- Representation (PB pictures, etc) and temporal prediction
- Motion and temporal prediction
- Loop filter
- Spatial prediction
- Transform
- Quantization
- VLC

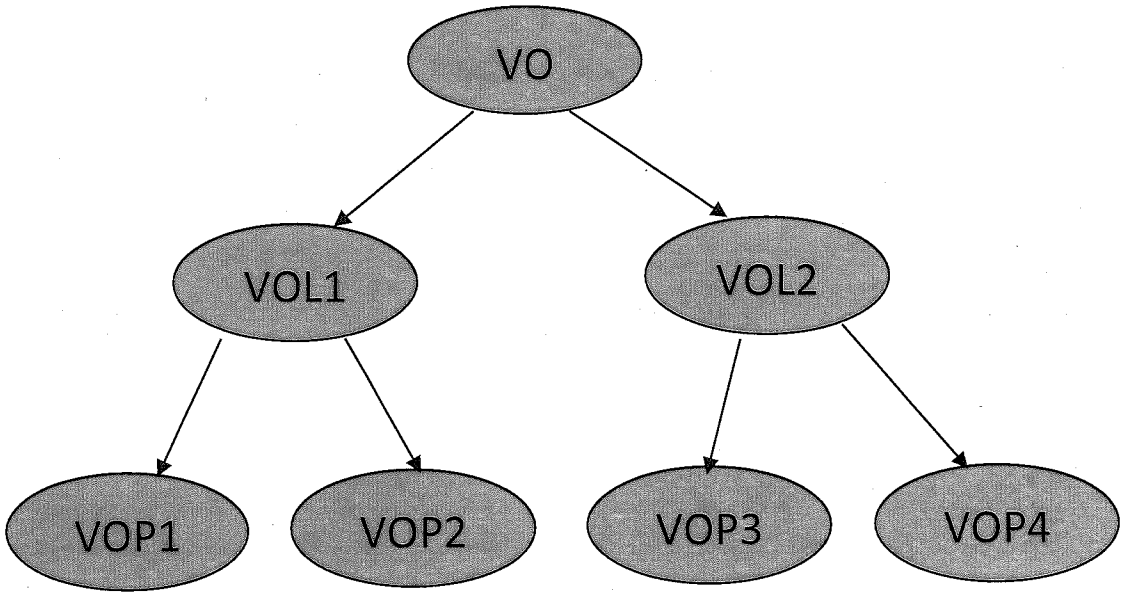
Evolution of the standards

- Representations of pictures
- Coding structure (picture types and prediction structures)
- Motion representations
- Loop filters
- Spatial prediction (intra-blocks and..)
- Transforms
- Quantizers
- Variable length coding

Representation: Picture formats

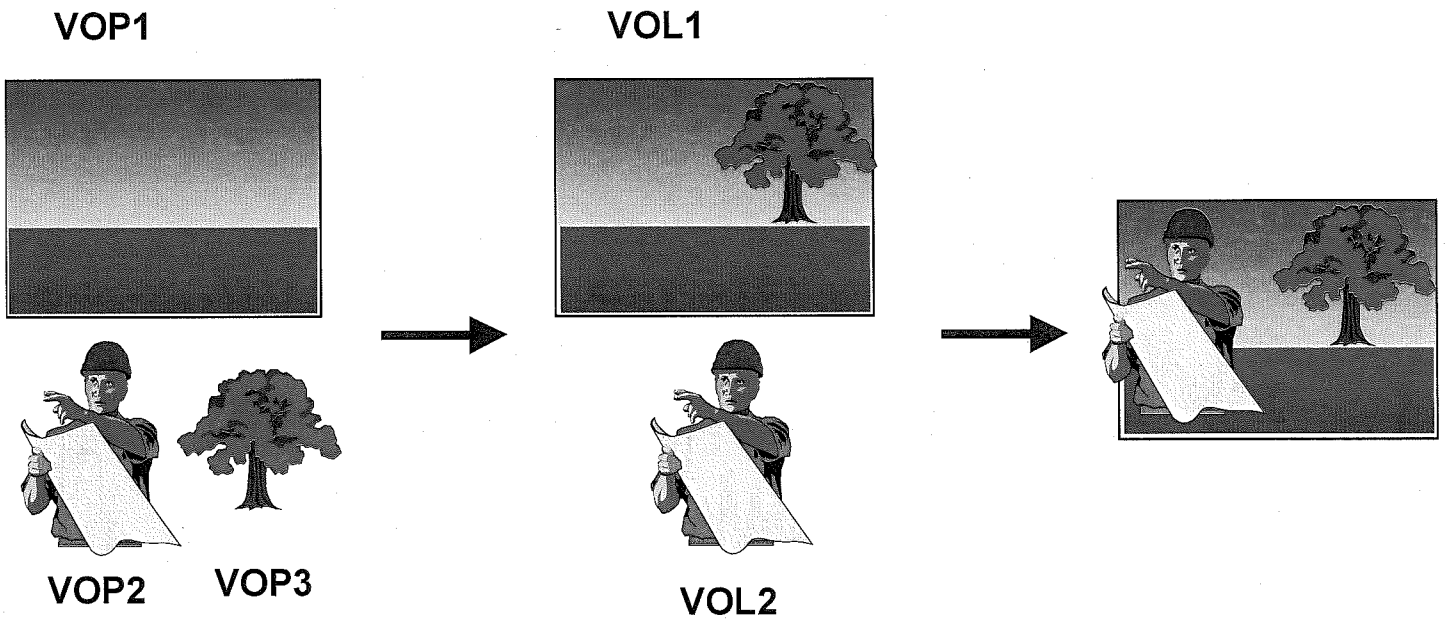
- Progressive pictures
  - H.261: only CIF (352*288) and QCIF (176*144)
  - MPEG-1:Progressive using SIF video format (352*240p30 or 352*288p25)
  - All other standards
- Interlaced pictures
  - MPEG-2 and beyond (but not HEVC!)
  - MPEG-2 introduces concept of field pictures
  - 4:2:0 format is modified to shift chroma samples
- Video Object Planes (VOP) == Objects
  - MPEG-4 only
  - Allows interactive editing of objects

Object Description Hierarchy in MPEG-4



VO: video object
VOL: video object layer
(can be different parts of a VO or different rate/resolution representation of a VOL)
VOP: video object plane

Example of Scene Composition



The decoder can compose a scene by including different VOPs in a VOL

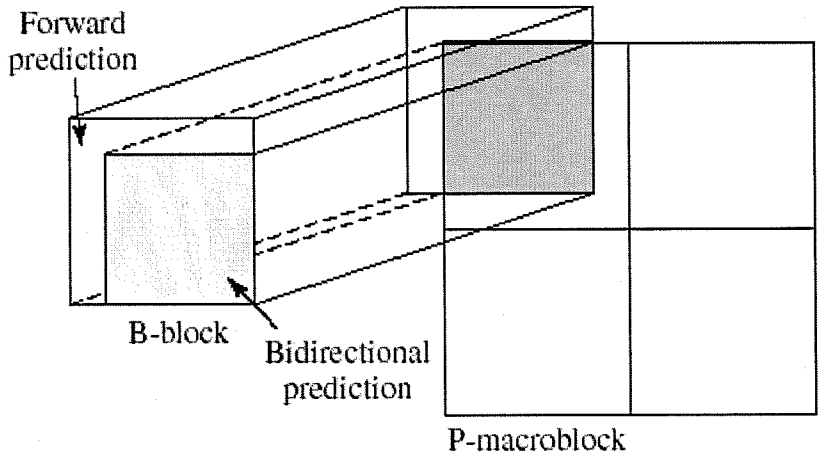
Object-Based Coding

- Entire scene is decomposed into multiple objects
  - Object segmentation is the most difficult task!
  - But this does **not** need to be standardized
- Each object is specified by its shape, motion, and texture (color)
  - Shape and texture both changes in time (specified by motion)
- MPEG-4 assumes the encoder has a segmentation map available, specifies how to **decode** (not encode!) shape, motion and texture

Representation: coding structure

- Motion-compensated predictive (P) frames
  - H.261 and all later standards
- Bidirectional (B) motion-compensated frames
  - Introduced in MPEG-1 for improved efficiency
  - Not included in H.263 (baseline) due to increased delay
  - Re-included in all subsequent standards (H.263+)
- PB-frames
  - Combination of P and B, appears only in H.263
- Field pictures and field prediction
  - MPEG-2 and beyond (until HEVC!)

H.263: PB-Picture Mode

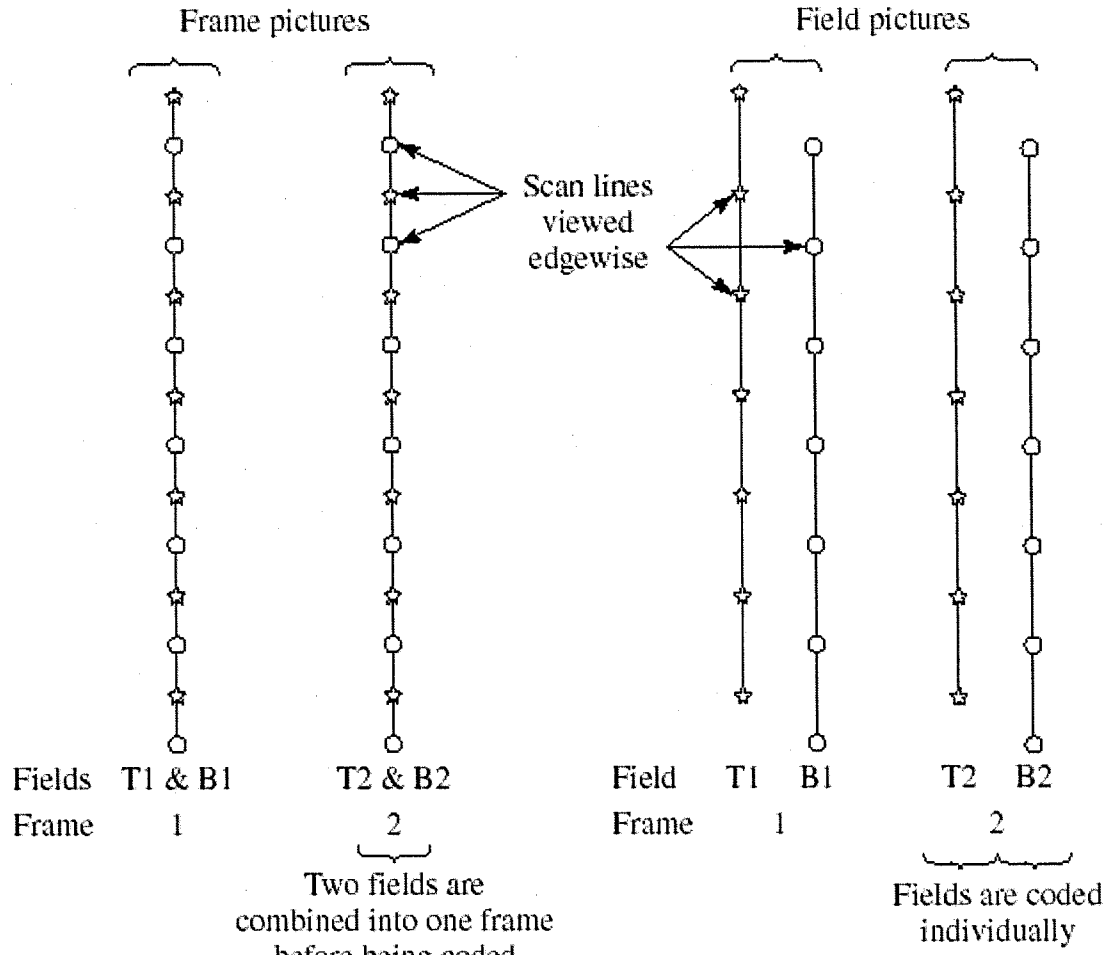


PB-picture mode codes two pictures as a group. The second picture (P) is coded first, then the first picture (B) is coded using both the P-picture and the previously coded picture. This is to avoid the reordering of pictures required in the normal B-mode. But it still requires additional coding delay than P-frames only.

In a B-block, forward prediction (predicted from the previous frame) can be used for all pixels; backward prediction (from the future frame) is only used for those pels that the backward motion vector aligns with pels of the current MB. Pixels in the "white area" use only forward prediction.

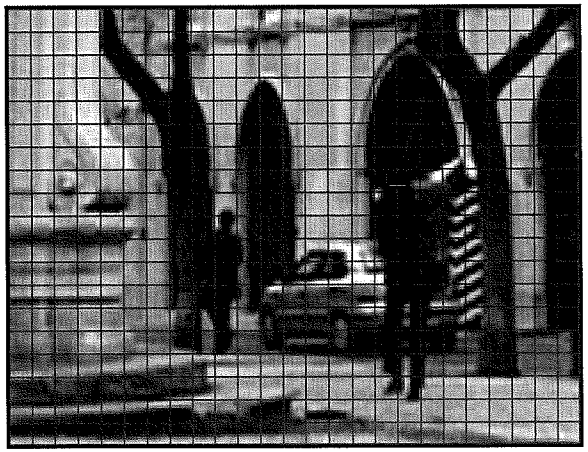
An improved PB-frame mode was defined in H.263+, that removes the previous restriction.

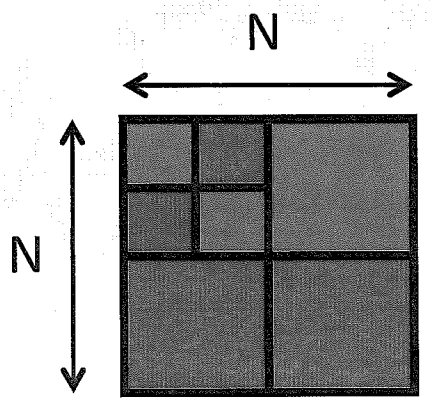
MPEG-2: Frame vs. Field Picture



Larger Coding Blocks

- Each frame is broken up into blocks
- Large block sizes reduce signaling overhead
- In H.264/AVC, macroblock is always 16x16 pixels
  - Each macroblock is either inter or intra coded
- In HEVC, Coding Tree Unit (CTU) can have up to 64x64 pixels
  - CTU can have a combination of inter and intra coded blocks

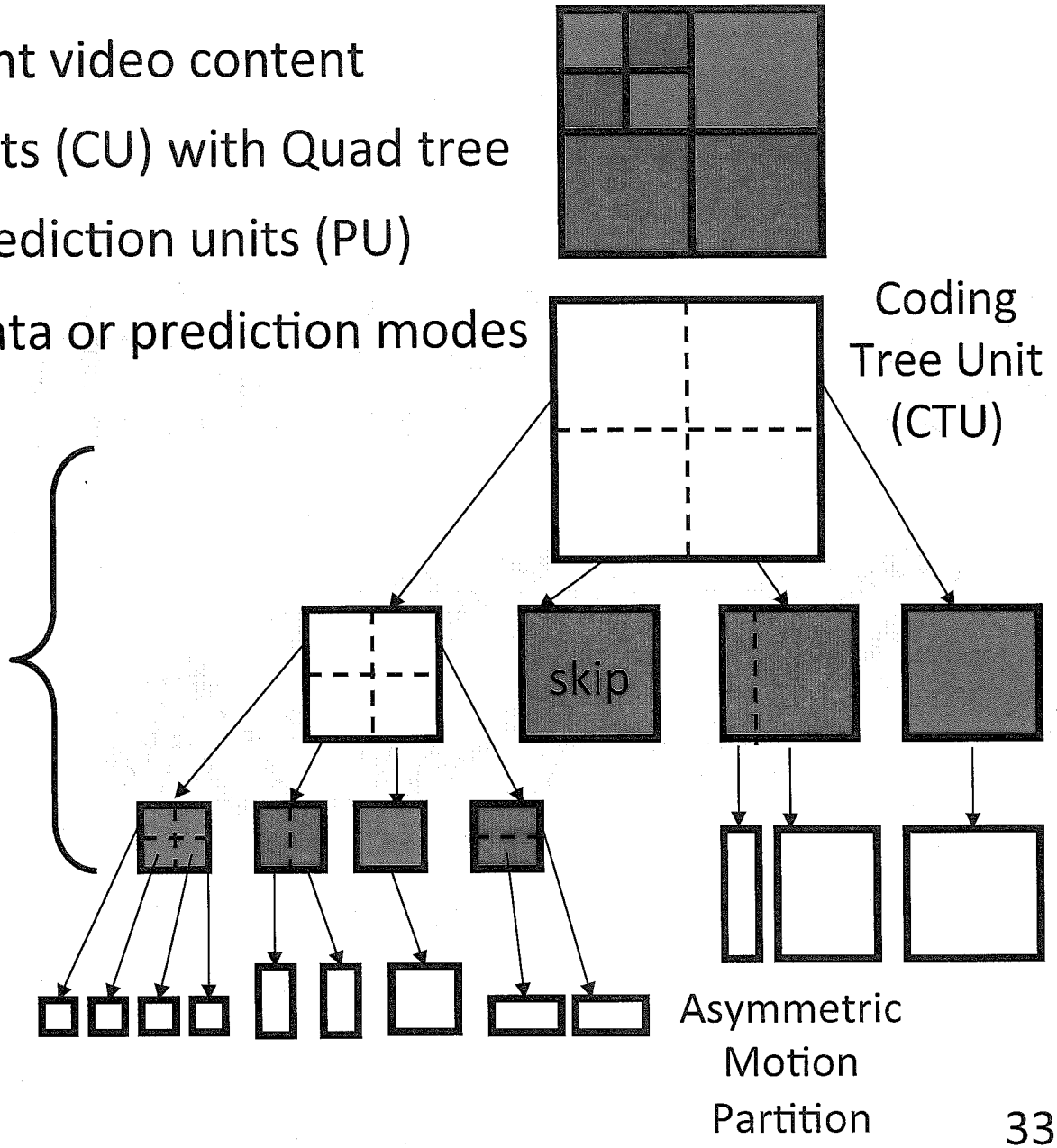




N=16, 32, or 64

Flexible Coding Block Structure

- Better adaptation to different video content
- CTU divided into Coding Units (CU) with Quad tree
- Coding units divided into prediction units (PU)
- PU have different motion data or prediction modes



Motion Representation and coding

- Methods for generating the MVs are not specified in the standard
- H.261
  - Forward prediction only
  - Integer accuracy only, range [-16,16]
- H.263
  - Forward and backward prediction (B and PB-frames)
  - Half-pixel motion accuracy; bilinear interpolation filter; range [-31.5,31]
  - Optional: unrestricted motion (across picture boundary)
  - Optional: Overlapped block motion estimation
  - Optional: 4 motion vectors per Macroblock

Overlapped Block Motion Compensation (OBMC)

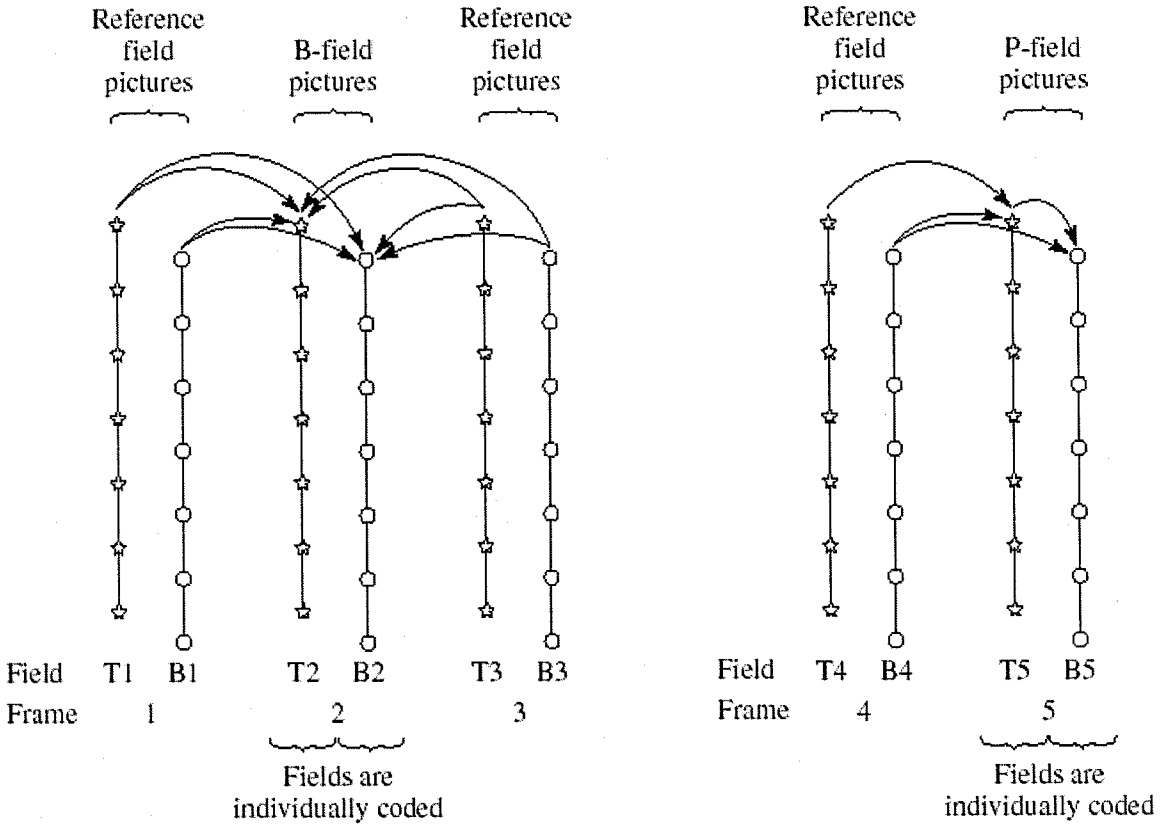
- Conventional block motion compensation
  - One best matching block is found from a reference frame
  - The current block is replaced by the best matching block
- OBMC
  - Each pixel in the current block is predicted by a weighted average of several corresponding pixels in the reference frame
  - The corresponding pixels are determined by the MVs of the current as well as adjacent MBs
  - The weights for each corresponding pixel depends on the expected accuracy of the associated MV

Motion, continued

- MPEG-1
  - Using more advanced motion compensation
    - Half-pel accuracy motion estimation, range [-64,64]
  - Using bi-directional temporal prediction
- MPEG-2
  - Field prediction for field pictures
  - Field prediction for frame pictures
  - Dual prime for P-pictures (One MV for two fields)
  - 16x8 MC for field pictures

Field prediction for field pictures

- Each field is predicted individually from the reference fields
  - A P-field is predicted from one previous field
  - A B-field is predicted from two fields chosen from two reference pictures



Field Prediction for Frame Pictures

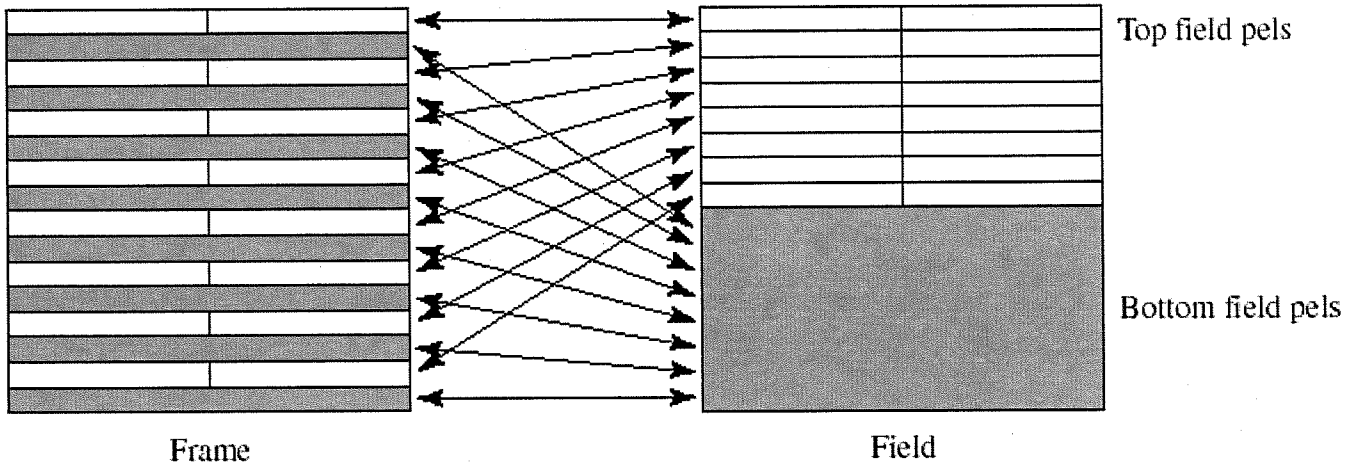


Figure 13.18 Field prediction for frame pictures: the MB to be predicted is split into top field pels and bottom field pels. Each 16 × 8 field block is predicted separately with its own motion vector (P-frame) or two motion vectors (B-frame).

Useful for rapid motion

MPEG-4 Prediction

- Quarter-pel motion estimation
- Four MVs and Unrestricted MVs (range [-2048,2048])
- Optional OBMC

- Sprite coding
  - Code a large background in the beginning of the sequence, plus affine mappings, which map parts of the background to the displayed scene at different time instances
  - Decoder can vary the mapping to zoom in/out, pan left/right

- Global motion compensation
  - Using 8-parameter projective mapping
  - Effective for sequences with large global motion

H.264 Motion Compensation

- Quarter-pel accuracy
- Variable block size
- Multiple reference frames
  - Generalized B-picture
- Weighted prediction (fade in, fade out, etc)

H.264 Variable Blocksize Motion Compensation

- Use variable size block-based motion compensation
  - 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4
  - H.263/MPEG4 use only 16x16 and 8x8

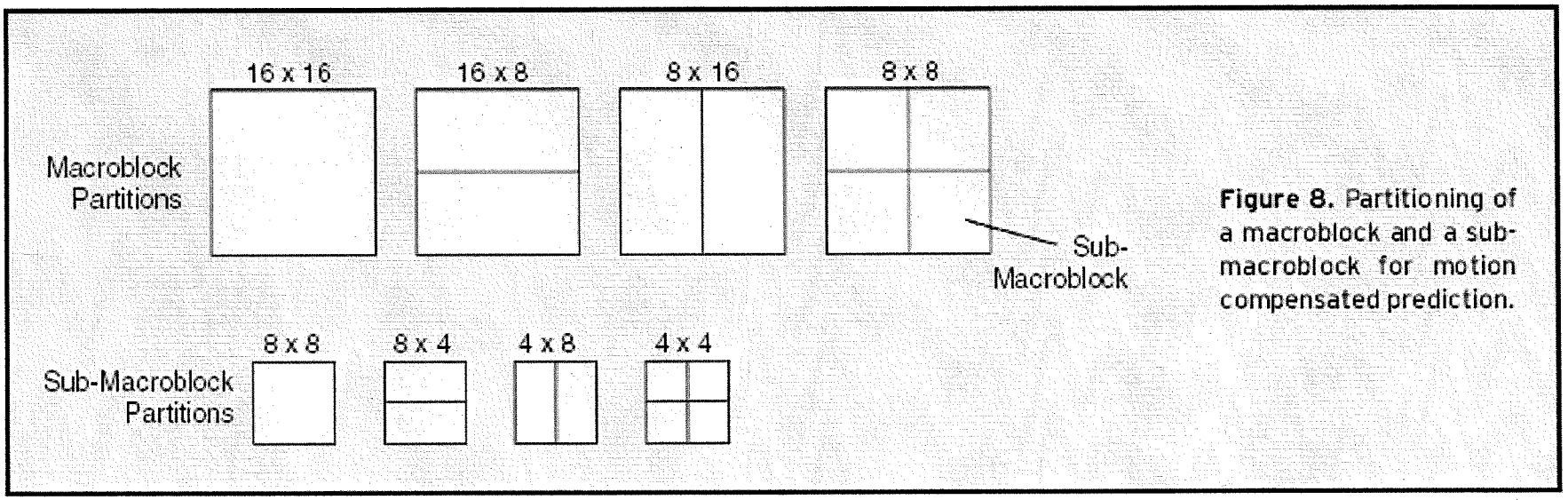


Figure 8. Partitioning of a macroblock and a sub-macroblock for motion compensated prediction.

From [Ostermann04]

H.264 Multiple Reference Frames for Motion Compensation

- Can use one or two from several possible reference frames
- When two reference frames are used, arbitrary weights can be used to combine them – Generalized B-picture

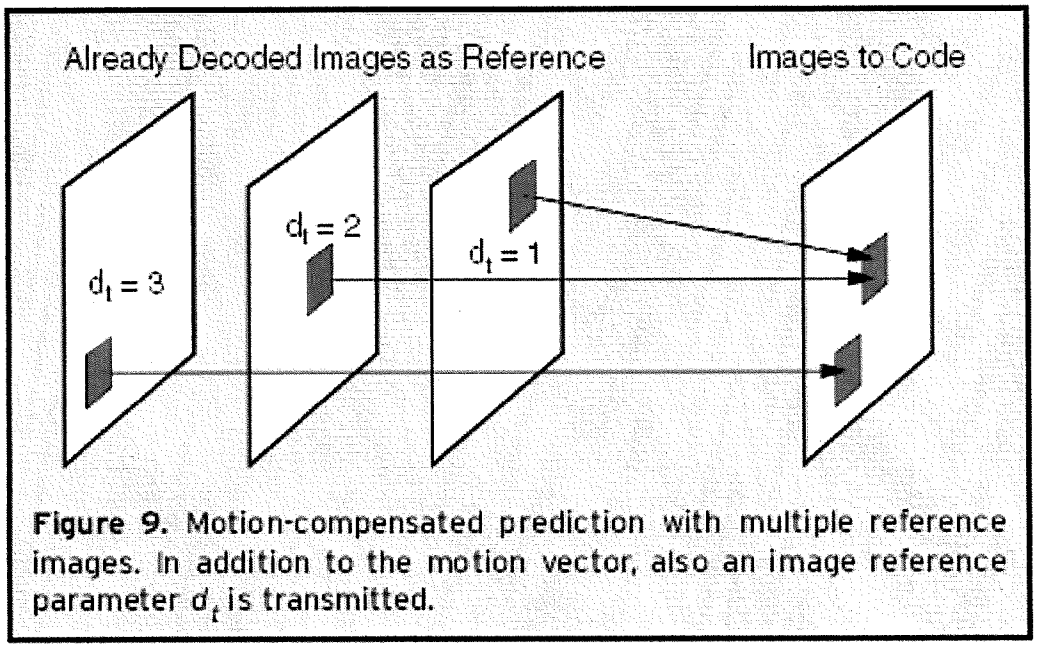
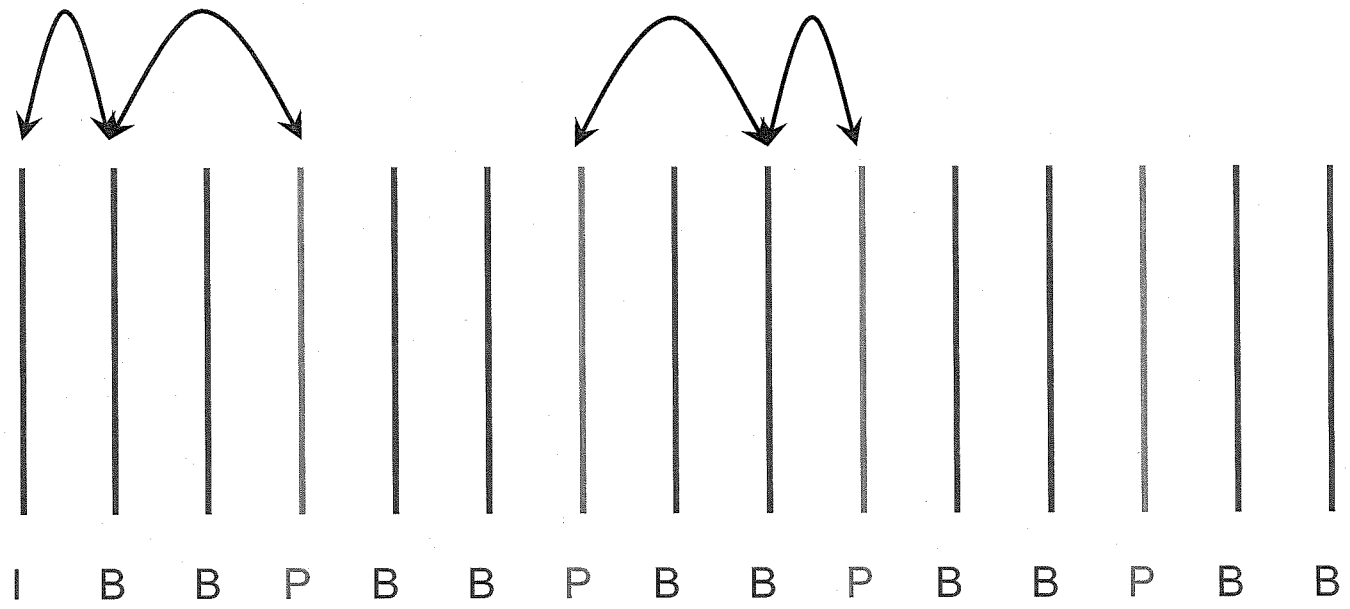


Figure 9. Motion-compensated prediction with multiple reference images. In addition to the motion vector, also an image reference parameter d_t is transmitted.

From [Ostermann04]

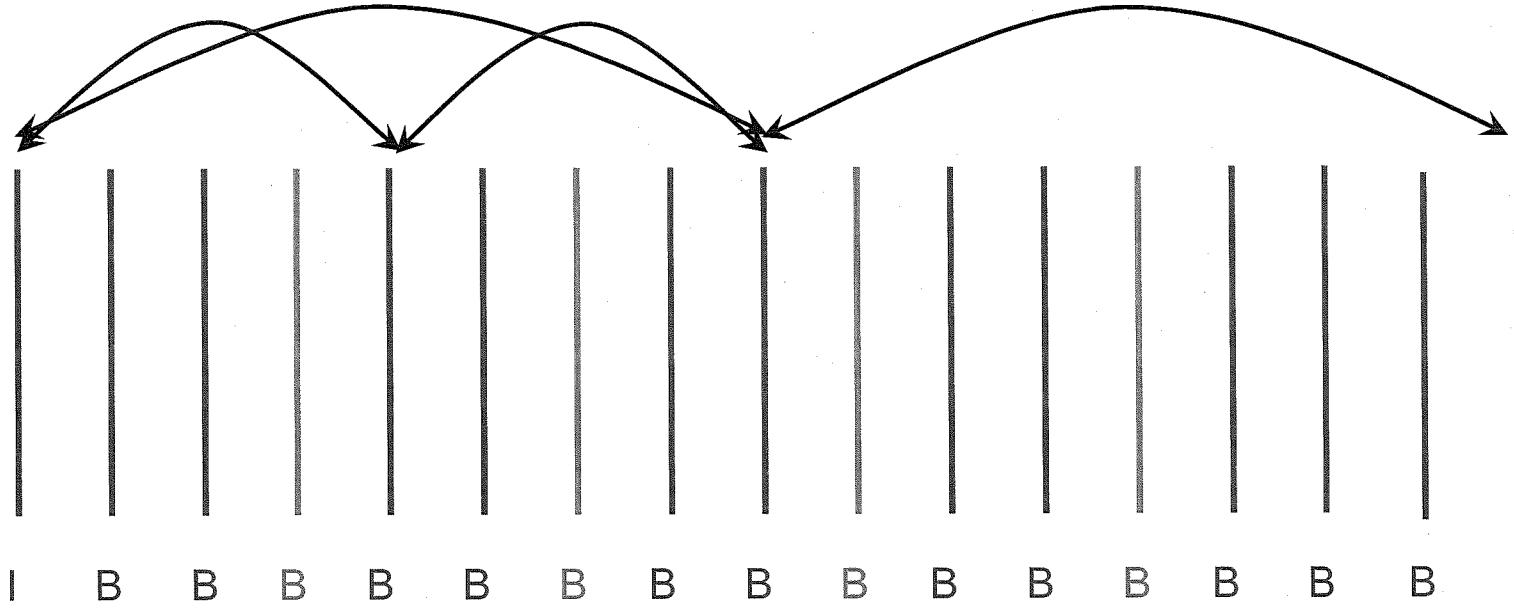
MPEG-1 Group of Pictures



Display order:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Encoding order:	0	2	3	1	5	6	4	8	9	7	11	12	10	14	15	13

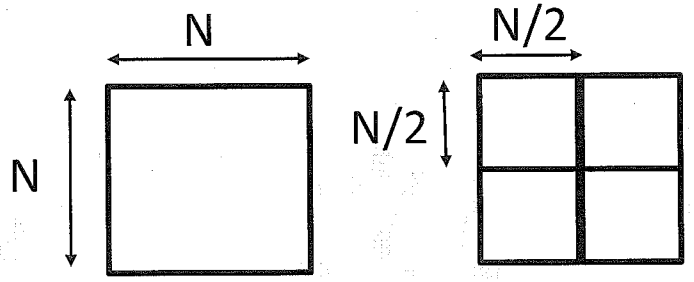
H.264 Generalized GOP

In H.264, B frames can be used for prediction



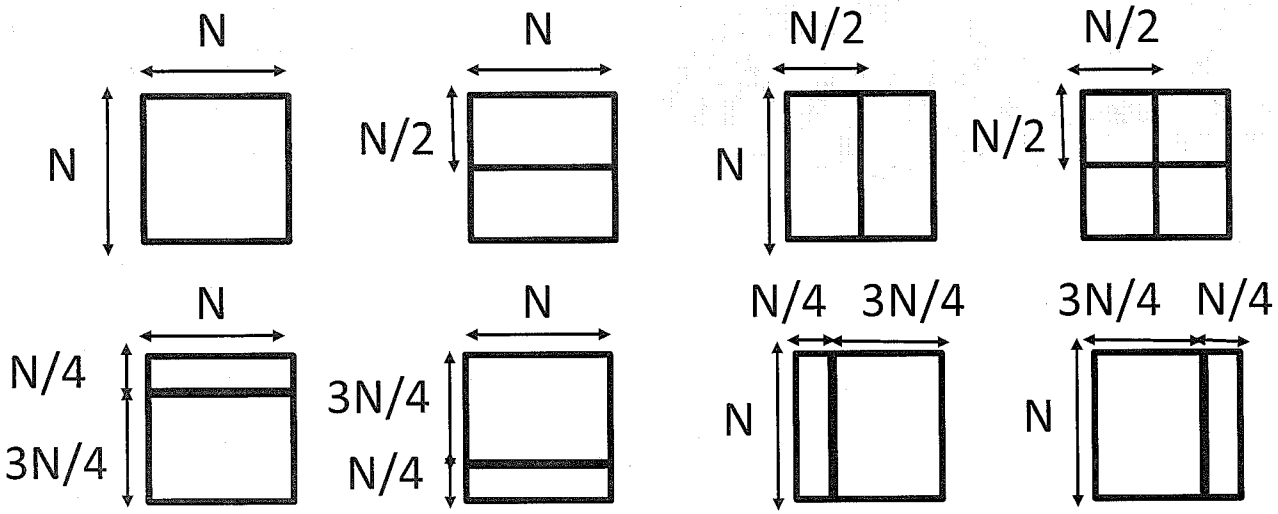
Prediction Units

- Intra-Coded CU can only be divided into square partition units
  - For a CU, make decision to split into four PU (8x8 CUs only) or single PU



Two methods of partitioning for intra-coded CU

- Inter-Coded CU can be divide into square and non-square PU as long as one side is at least 4 pixels wide (note: no 4x4 PU)

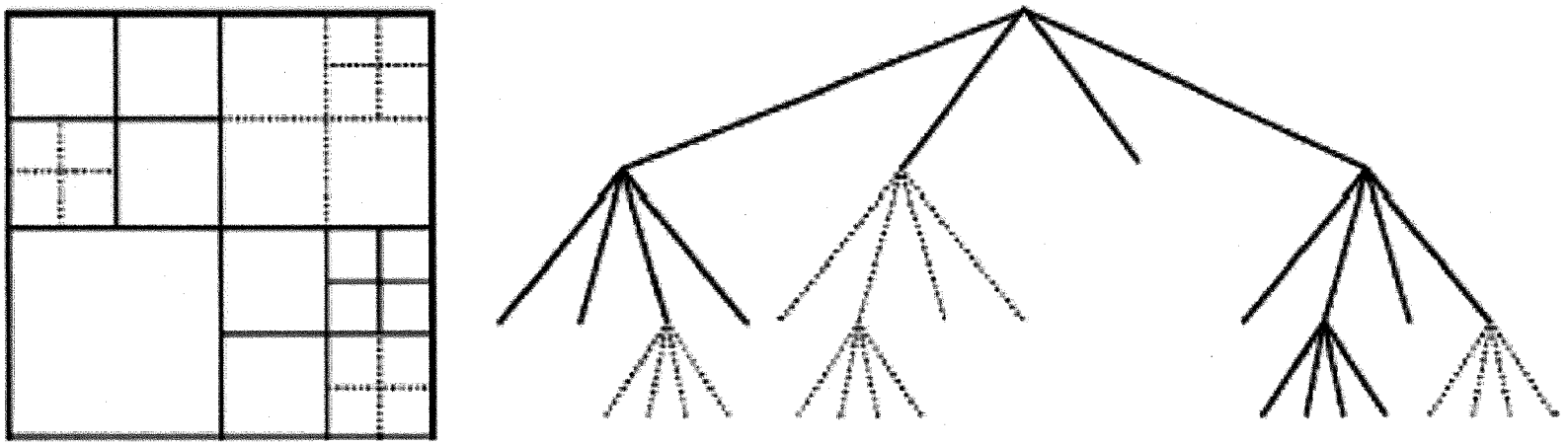


Eight methods of partitioning for inter-coded CU

Tree-Structured Partitioning into Transform Blocks and Units

- For residual coding, a CB can be recursively partitioned into transform blocks.
- The partitioning is signaled by a residual quadtree.

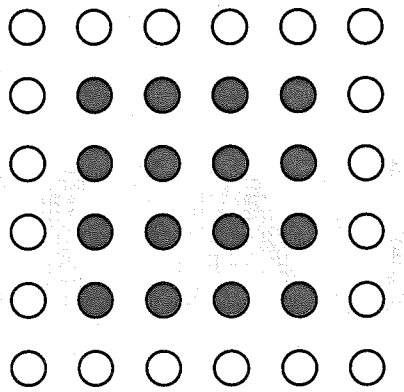
Tree-Structured Partitioning into Transform Blocks and Units



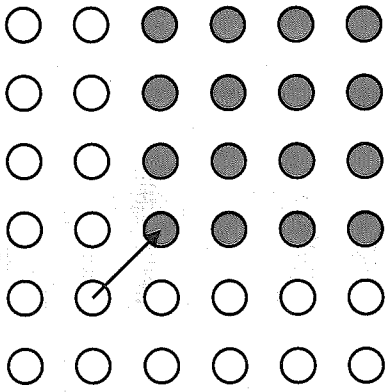
- Subdivision of a CTB into CBs and TBs.
- Solid lines: CB boundaries, dotted lines: TB boundaries

Inter Prediction

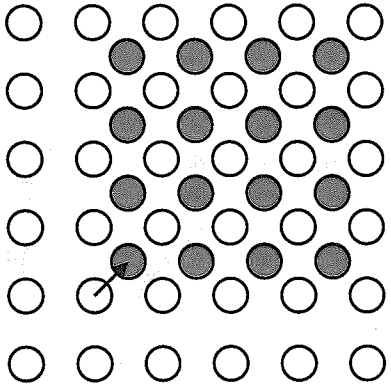
- Motion vectors can have up to ¼ pixel accuracy (interpolation required)



4x4 block in current frame



Reference block in previous frame
Vector (1, -1)



Reference block in previous frame
Vector (0.5, -0.5)

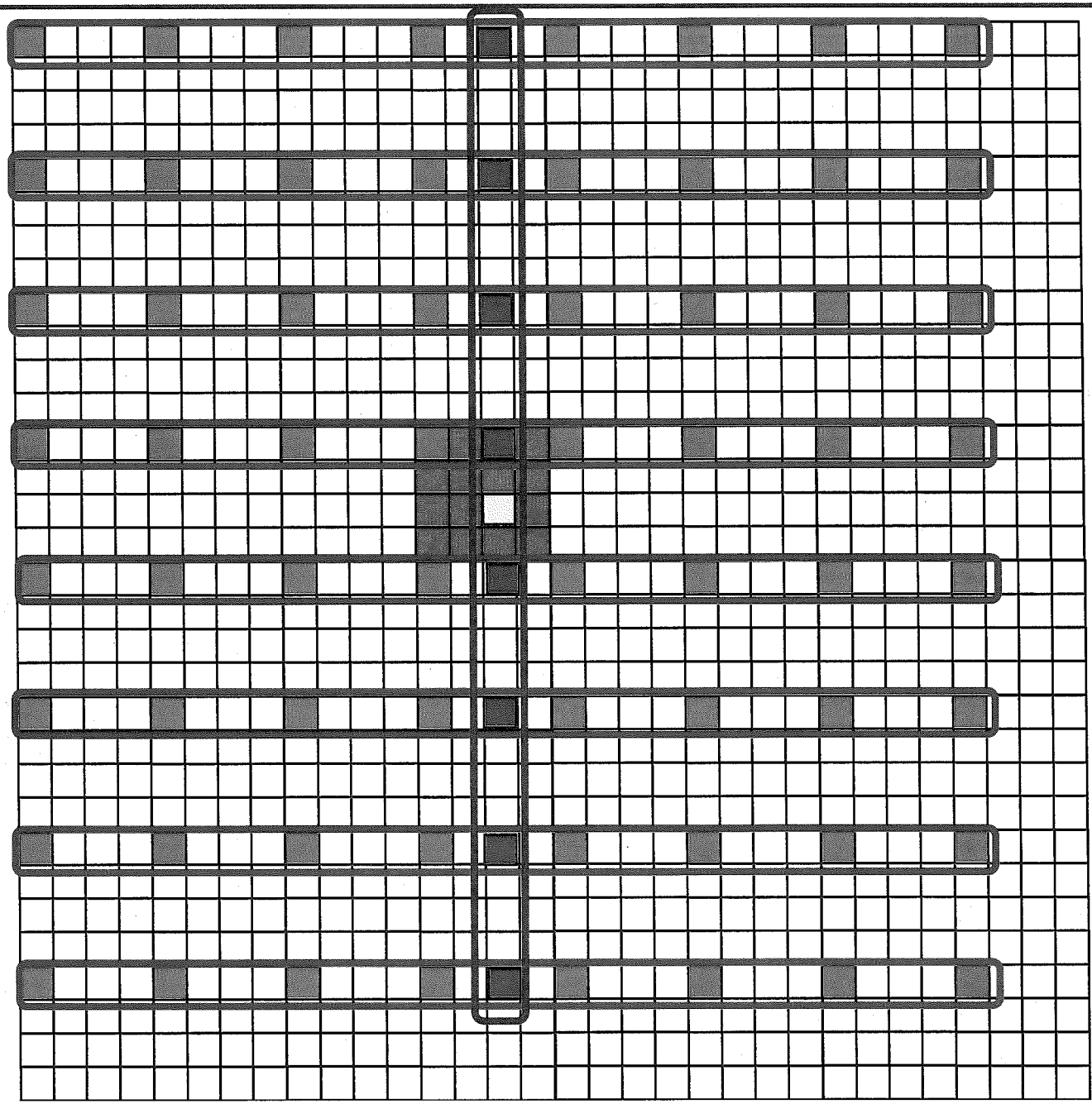
- In H.264/AVC, luma uses 6-tap filter, and chroma uses bilinear filter
- In HEVC, luma uses 8/7-tap and chroma uses 4-tap
  - Different coefficients for ¼ and ½ positions
- Restricted prediction on small PU sizes

Interpolation Filter

Require integer pixels (highlighted in red) to interpolate fractional pixels (highlighted in blue)

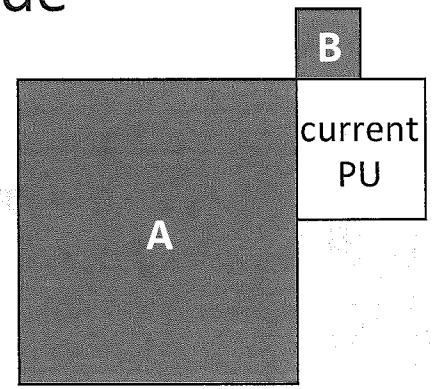
To interpolate NxN pixels requires up to (N+7)x(N+7) reference pixels

Use 1-D filters (order matters for greater than 8-bit video)



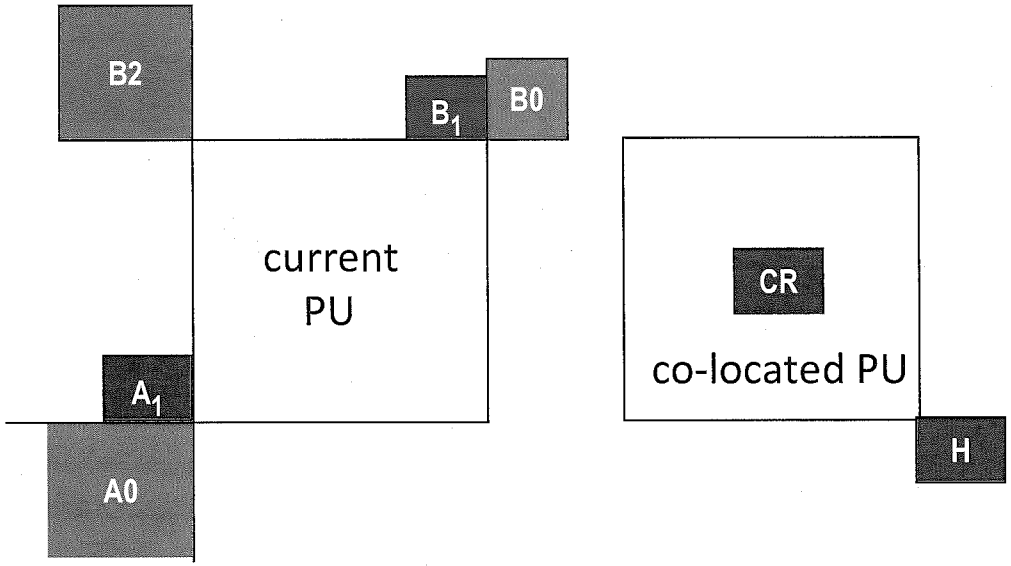
Mode Coding

- Predict modes from neighbors to reduce syntax element bits
  - Intra Prediction Mode



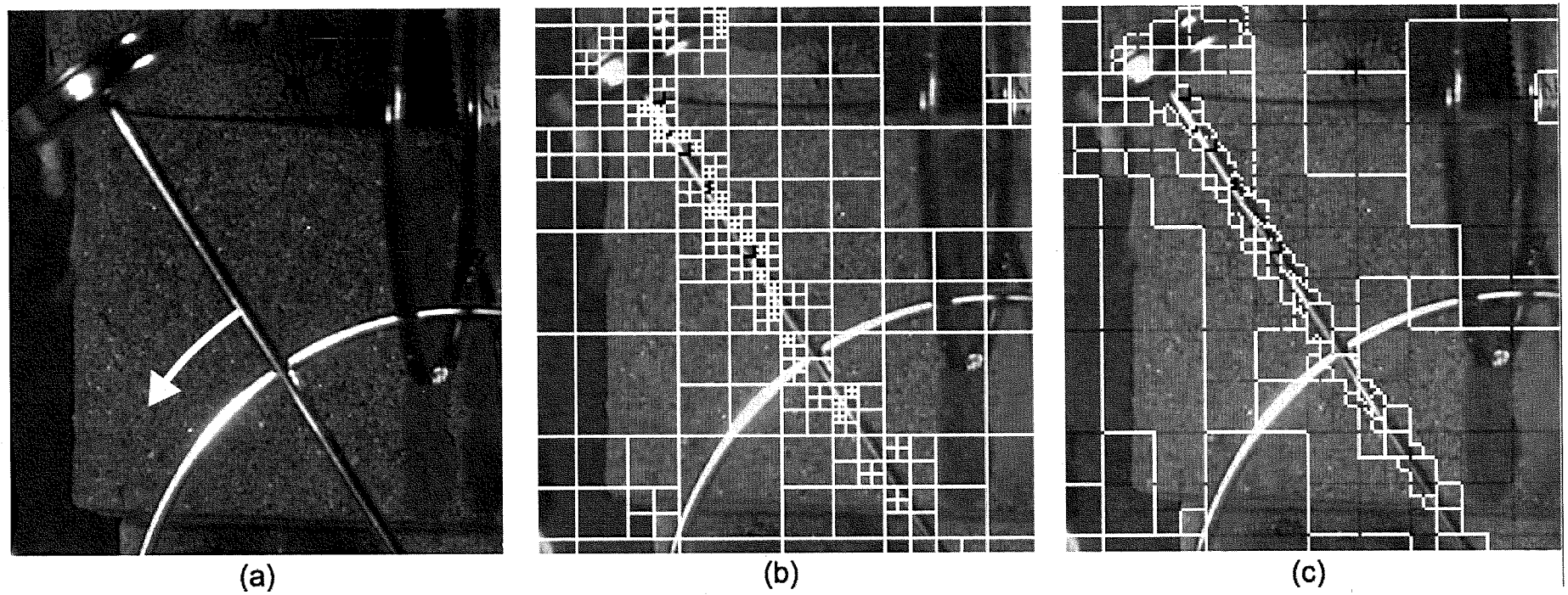
3 candidates

  - Advance Motion Vector Prediction (AMVP), Merge/Skip Mode



2 to 5 candidates

Merge Mode



Moving Object

Without Merge
(many extra motion parameters)

With Merge

B. Bross et al., "Inter Prediction in HEVC," *High Efficiency Video Coding (HEVC): Algorithms and Architectures*, Springer, 2014.

AMVP, Merge, Skip Mode

	AMVP	Merge	Skip
Syntax elements	mvp_l0_flag, mvp_l1_flag	merge_flag, merge_idx	cu_skip_flag, merge_idx
Use of neighbors candidates	Predict motion vector	Copy motion data (motion vector, reference index, direction)	Copy motion data (motion vector, reference index, direction); no residual
Number of Candidates	Up to 2	Up to 5 (signaled in slice header)	
Spatial	Up to 2 of 5 (scaling if reference index different)	Up to 4 of 5 (no scaling, only redundancy check)	
Temporal	Up to 1 of 2 (if < 2 spatial candidates)	Up to 1 of 2 (always added to list if available)	
Additional	Zero motion vector (if < 2 spatial or temp candidates)	Bi-predictive candidates and zero motion vector	

Spatial prediction

- H.261
  - Motion vector prediction using previously encoded MV
- MPEG-1
  - DC coefficients coded predictively
- H.263
  - MV prediction using the median of three neighbors
  - Optional: Intra DC prediction (10-15% improvement)
- MPEG-4
  - DC prediction: can predict DC coefficient from *either* the previous block or the block above
  - AC prediction: can predict one column/row of AC coefficients from *either* the previous block or the block above

H.264 Intra prediction

Apply prediction to the entire 16*16 block, or
apply prediction separately to sixteen 4*4 blocks

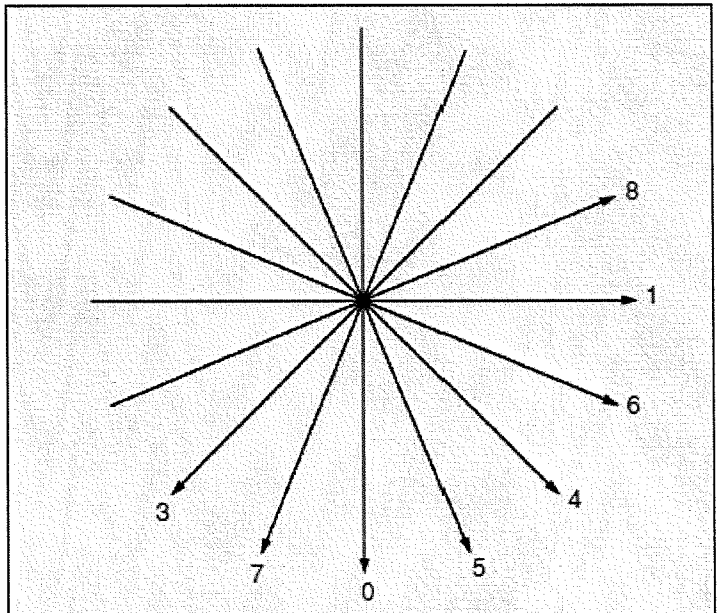
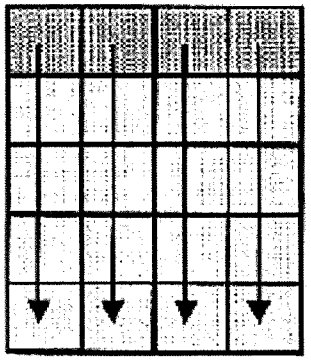


Figure 7. Possible prediction directions for INTRA_4×4 mode.

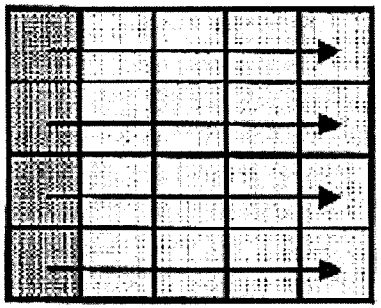
8 possible directions

Intra-Frame Prediction (h264)

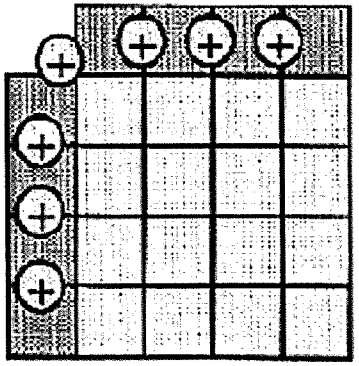
Mode 0 - Vertical



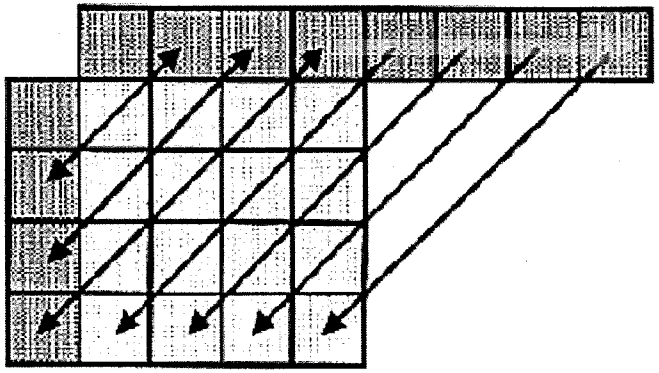
Mode 1 - Horizontal



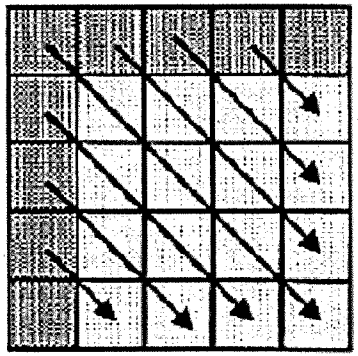
Mode 2 - DC



Mode 3 – Diagonal Down/Left



Mode 4 – Diagonal Down/Right



Across slice boundaries is not allowed.

H.264 Intra Prediction

- Instead of the simple DC coefficient prediction to exploit the correlation between nearby pixels in the same frame, more sophisticated spatial prediction is used, including INTRA_4x4 and INTRA_16x16

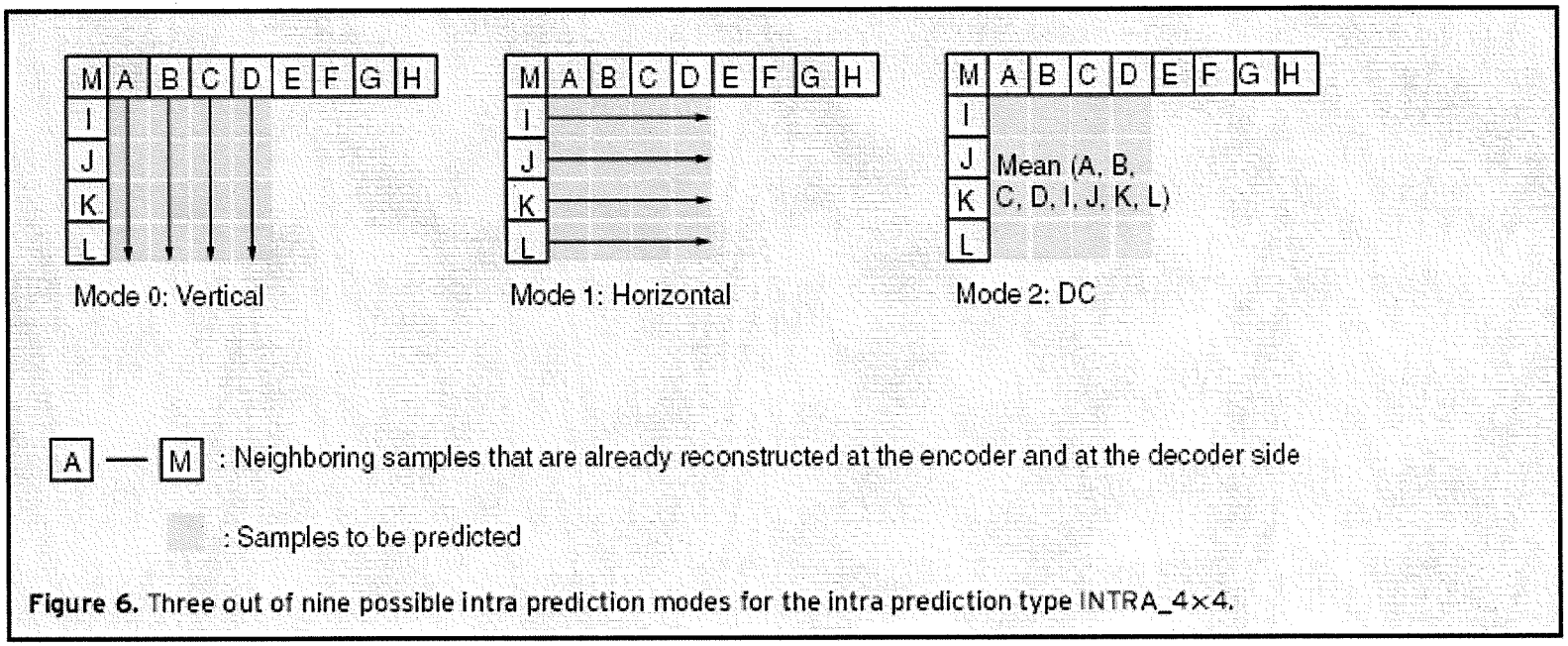
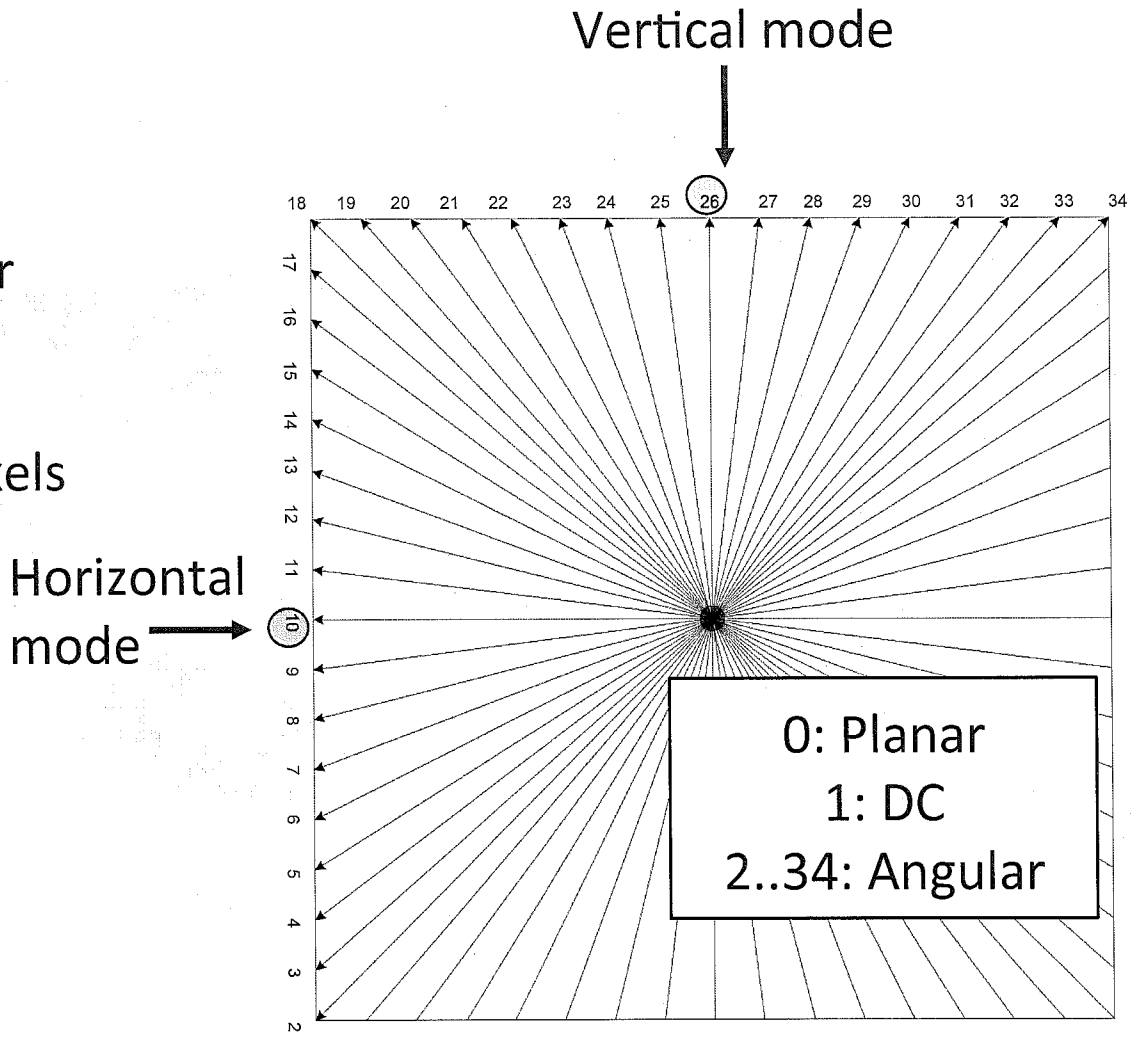


Figure 6. Three out of nine possible intra prediction modes for the intra prediction type INTRA_4×4.

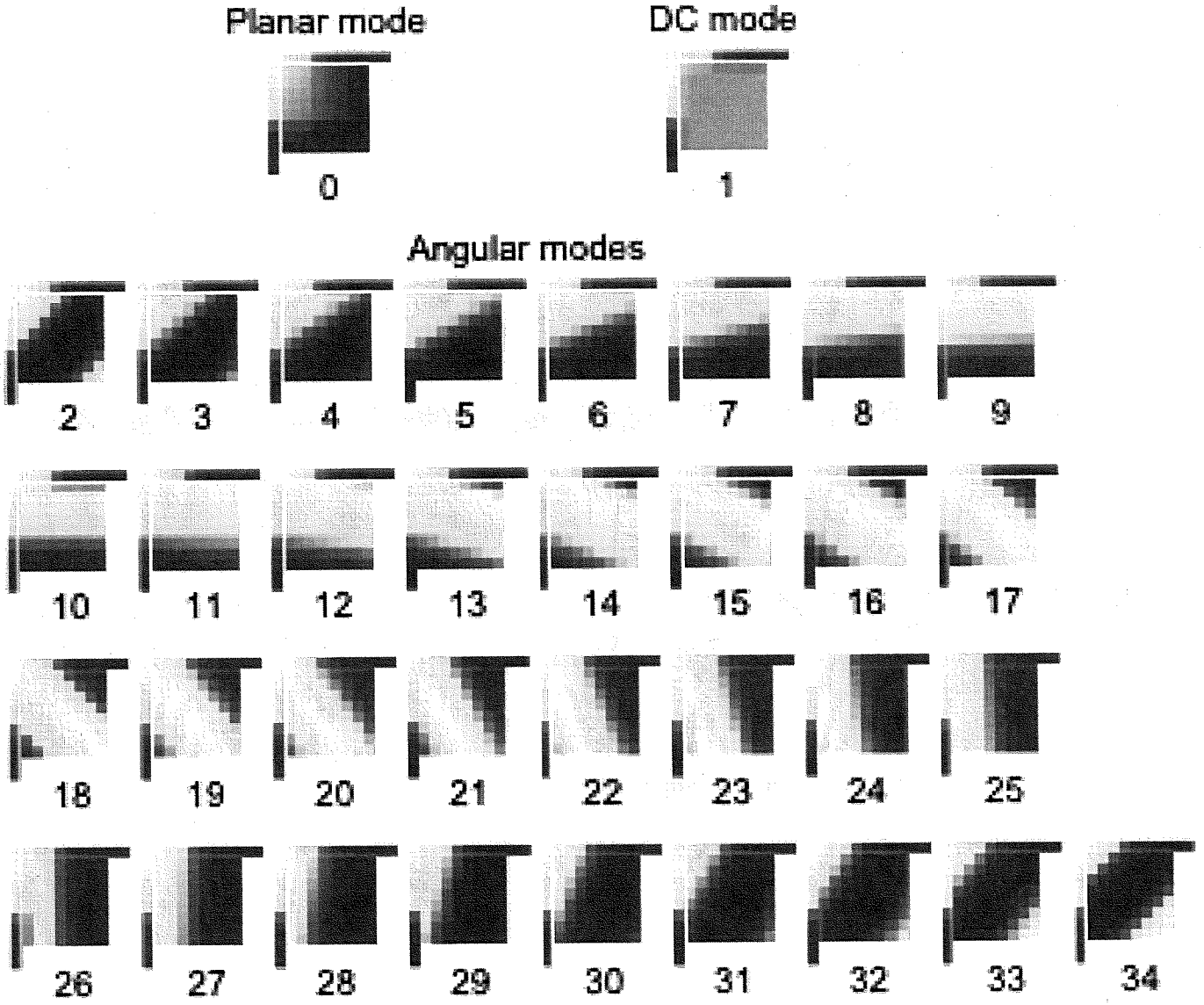
From [Ostermann04]

Intra Prediction

- H.264/AVC has 10 modes
  - angular (8 modes), DC, planar
- HEVC has 35 modes
  - angular (33 modes), DC, planar
- Angular prediction
  - Interpolate from reference pixels at locations based on angle
- DC
  - Constant value which is an average of neighboring pixels (reference samples)
- Planar
  - Average of horizontal and vertical prediction



Intra Prediction Modes



J. Lainema, W.-J. Han, "Intra Prediction in HEVC," *High Efficiency Video Coding (HEVC): Algorithms and Architectures*, Springer, 2014.

Intrapicture Prediction

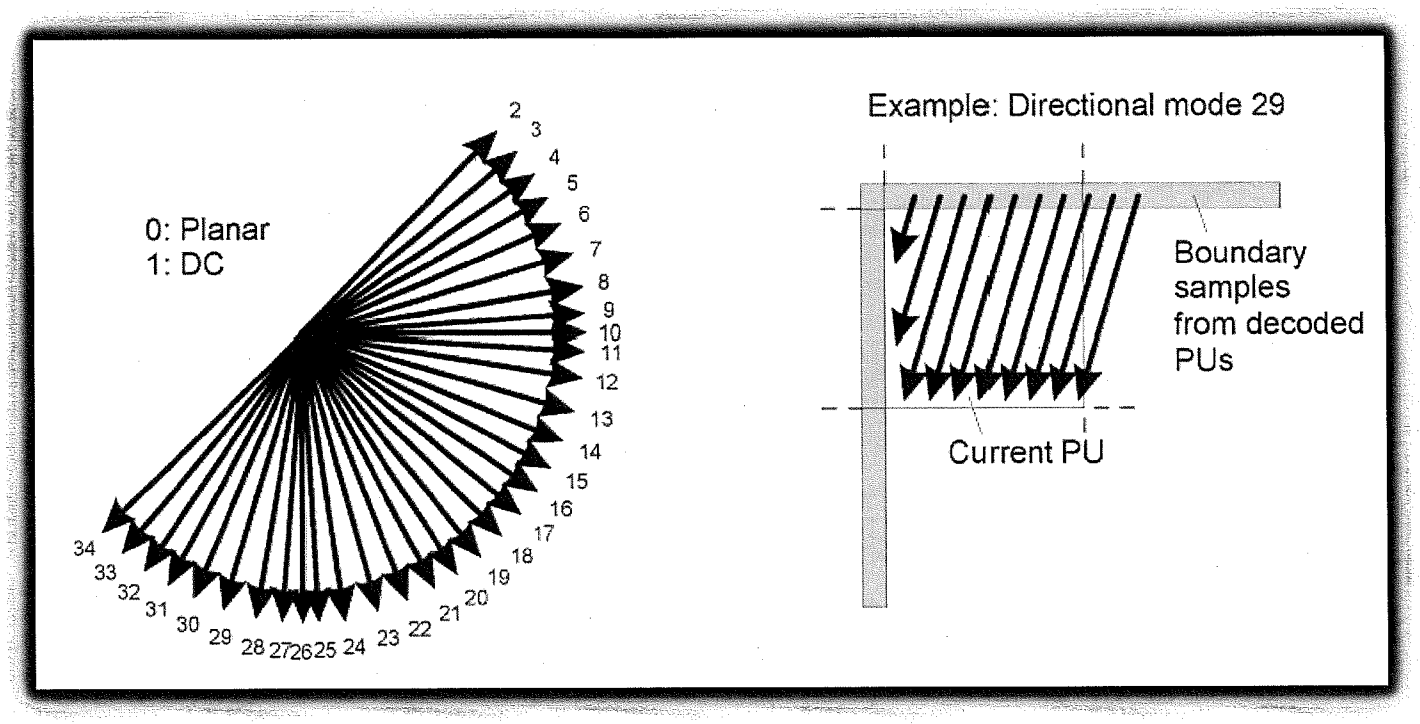


Fig. 6. Modes and directional orientations for intrapicture prediction

Transform

- 8x8 DCT
  - H.261
  - MPEG-1
  - H.263
  - MPEG-2
  - MPEG-4
- DCT is non-integer; the result depends on the implementation details

H.264 Integer Transform

- Smaller block size (4x4 or 2x2) can better represent boundaries of moving objects, and match prediction errors generated by smaller block size motion compensation
- Integer transform can be implemented more efficiently and no mismatch problem between encoder and decoder
- A few integers, approximates DCT, maintains orthogonality

Primary transform

$$H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad H_3 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Figure 10. Matrices H_1, H_2 and H_3 of the three different transforms applied in H.264/AVC.

From [Ostermann04]

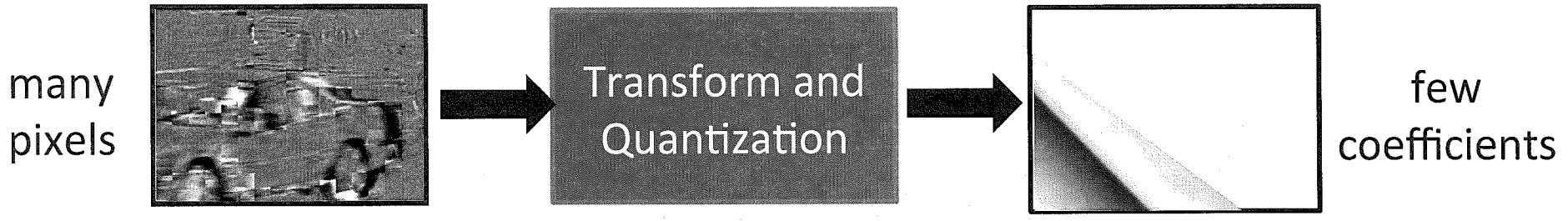
AVS Integer Transform

$$T_8 = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\ 10 & 4 & -4 & -10 & -10 & -4 & 4 & 10 \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\ 4 & -10 & 10 & -4 & -4 & 10 & -10 & 4 \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 \end{bmatrix}$$

Each row does not have the same length; this is taken into account in quantization

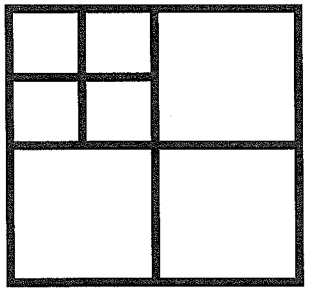
For AVS, this scaling occurs only in encoder, to make decoder simpler
For either H.264 or AVS, negligible loss in Transform Coding Gain

Large Transforms



- HEVC supports 4x4, 8x8, 16x16, 32x32 integer transforms
  - Two types of 4x4 transforms (IDST-based for Intra, IDCT-based for Inter); IDCT-based transform for 8x8, 16x16, 32x32 block sizes
  - Integer transform avoids encoder-decoder mismatch and drift caused by slightly different floating point representations.
  - Parallel friendly matrix multiplication/partial butterfly implementation
  - Transform size signaled using Residual Quad Tree
- Achieves 5 to 10% increase in coding efficiency
- Increased complexity compared to H.264/AVC
  - 8x more computations per coefficient
  - 16x larger transpose memory

Represent residual of CU with TU quad tree



M. Budagavi et al., "Core Transform Design in the High Efficiency Video Coding (HEVC) Standard," IEEE JSTSP, 2013

Transform, Scaling, and Quantization

- Alternative integer Transform
  - It is derived from a DST.
  - It is applied to only 4×4 luma residual blocks.
    - ➢For intrapicture prediction modes.
  - It is not much more computationally demanding than the 4×4 DCT-style transform.
  - It provides approximately 1% bit-rate reduction.