

فشرده‌سازی اطلاعات

۰۱-۷۰۲-۱۰-۱۴۰

بخش هشتم

قسمت دوم



دانشگاه شهید بهشتی
پژوهشکده‌ی فضای مجازی

بهار ۱۳۹۸

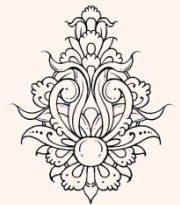
احمد محمودی ازناوه

فشرده‌سازی ویدئو

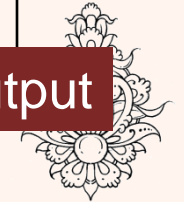
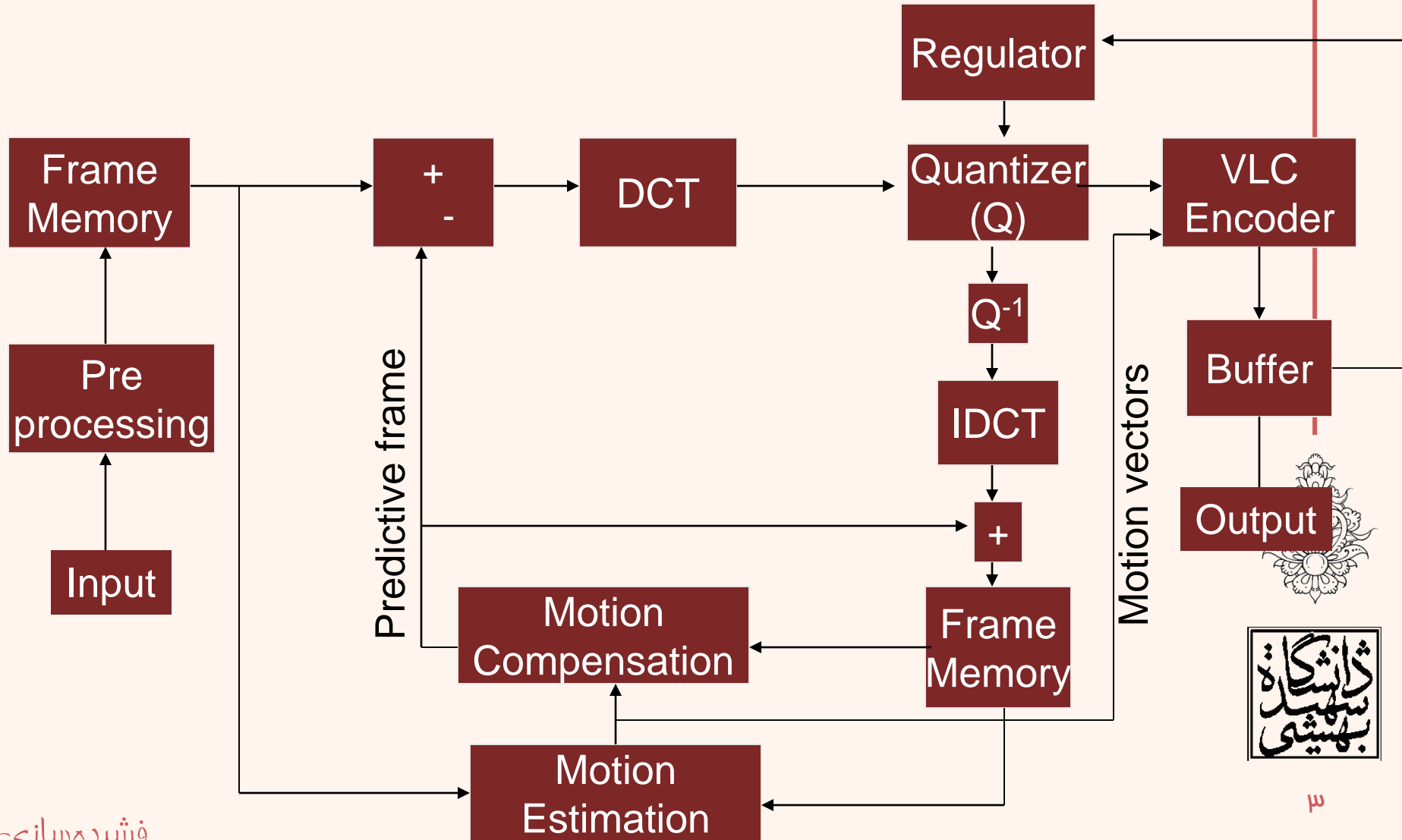
H.264

فهرست مطالب

- مقدمه‌ای بر بردارهای حرکت
- شمای کلی استاندارد H.264
- تخمین بردار حرکت
- پیش‌بینی بردار حرکت
- پیش‌بینی Intra
- تبدیل کسینوسی گسسته‌ی صحیح
- تبدیل هادامار
- نحوه‌ی محاسبات
- استاندارد HEVC



ساختار کدگذار MPEG2

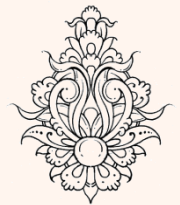


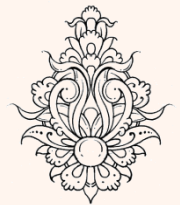
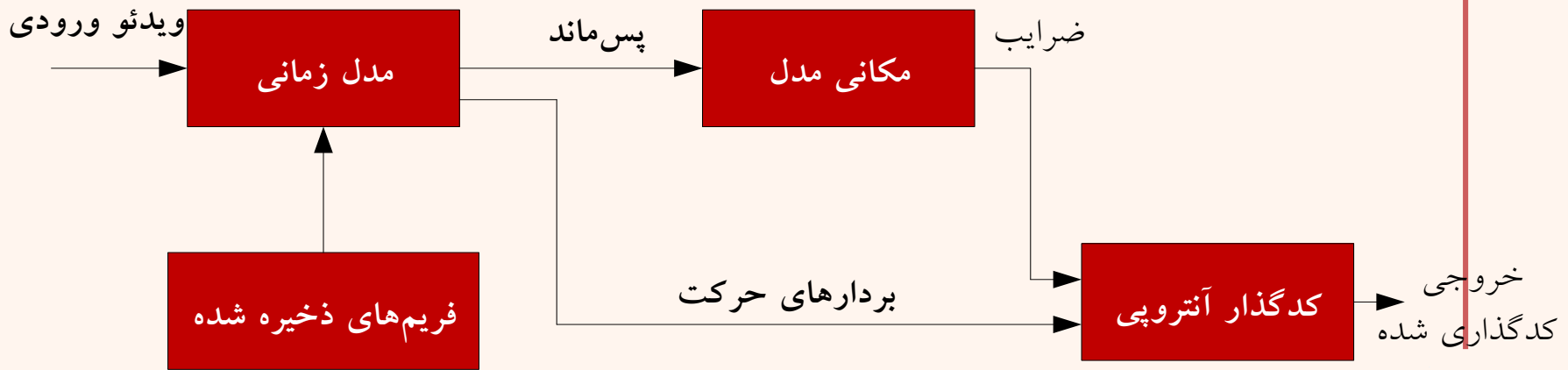
مفهوم فشردگی در یادآوری



Temporal model

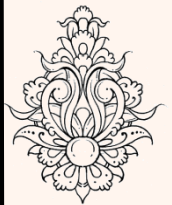
Spatial model





استانداردها

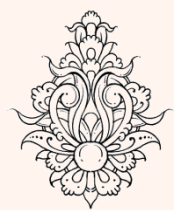
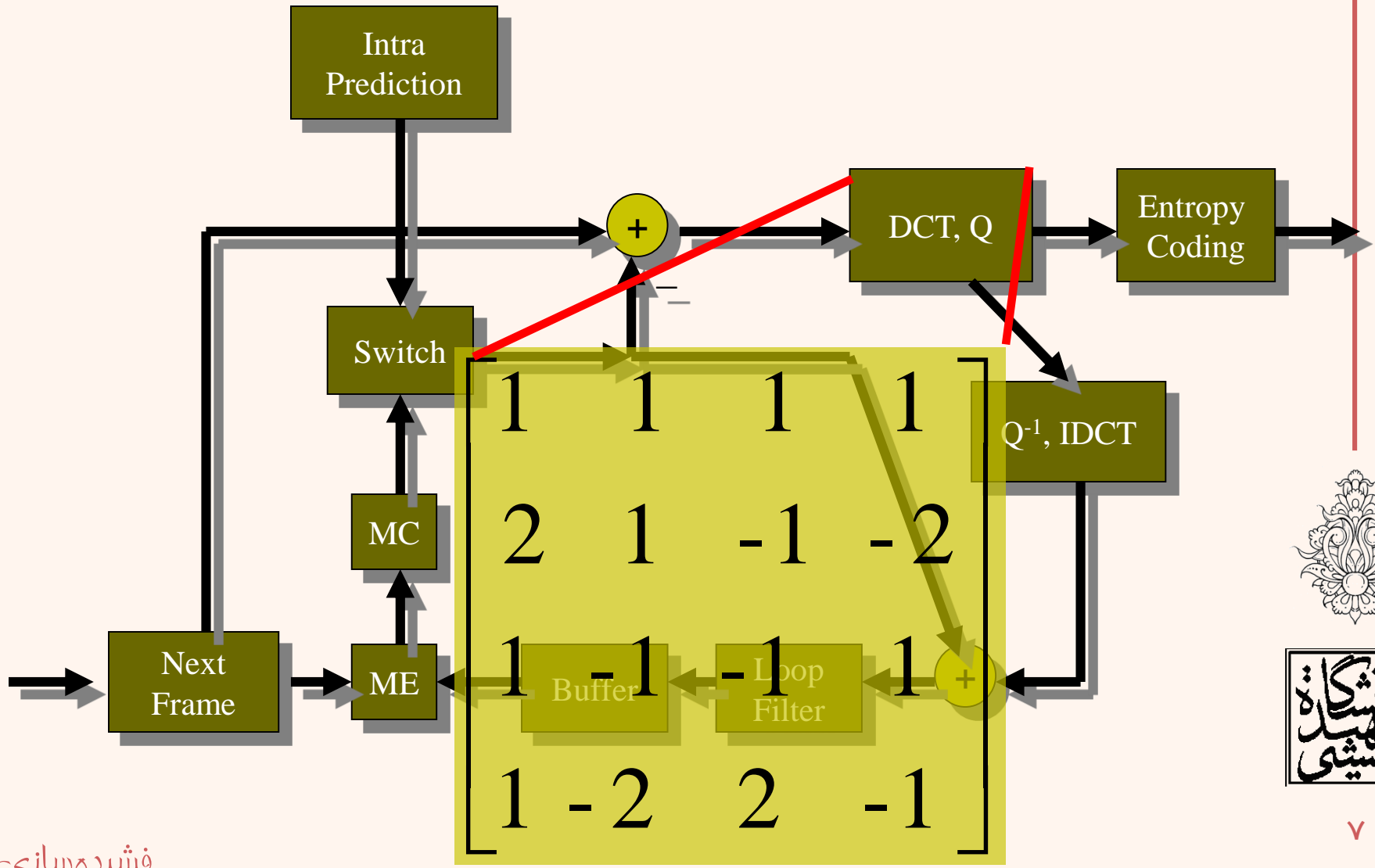
Standard	Main Applications	Year
JPEG, JPEG2000	Image	1992-1999, 2000
JBIG	Fax	1995-2000
H.261	Video Conferencing	1990
H.262, H.262+	DTV, SDTV	1995, 2000
H.263, H.263++	Videophone	1998, 2000
MPEG-1	Video CD	1992
MPEG-2	DTV, SDTV, HDTV, DVD	1995
MPEG-4	Interactive video	2000
MPEG-7	Multimedia Content description Interface	2001
MPEG-21	Multimedia Framework	2002
H.264/MPEG-4 part 10	Advanced Video Coding	2003
	Fidelity Range Extensions (High profile), Studio editing, Post processing, Digital cinema	2004 August
HEVC / H.265	The initial version of the H.265/HEVC standard was ratified in January, 2013	



تازشگاه
سپهبد
بهشتی

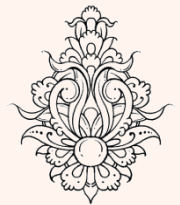
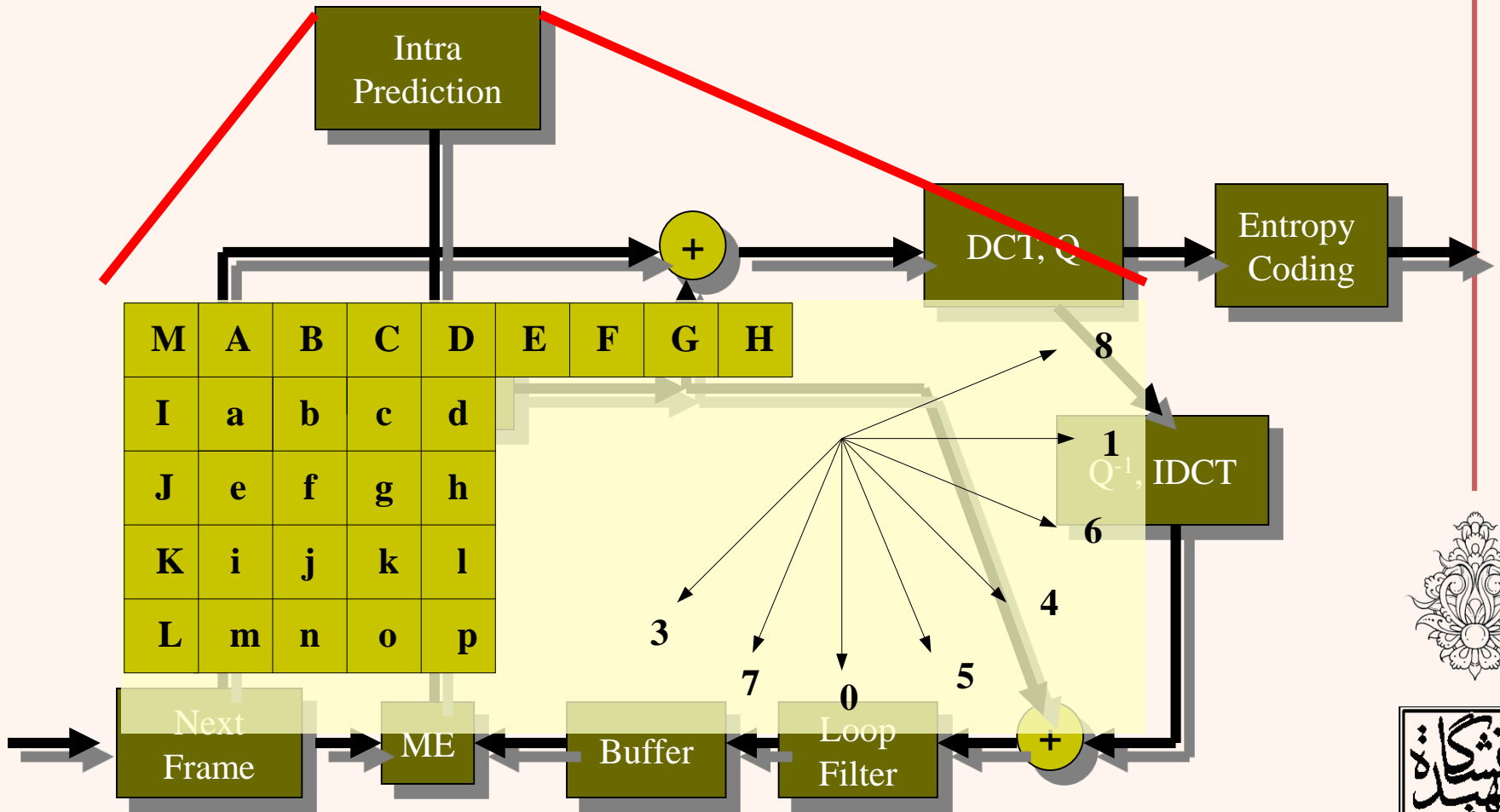
H.264 جزئیات بهبود صورت گرفته در استاندارد

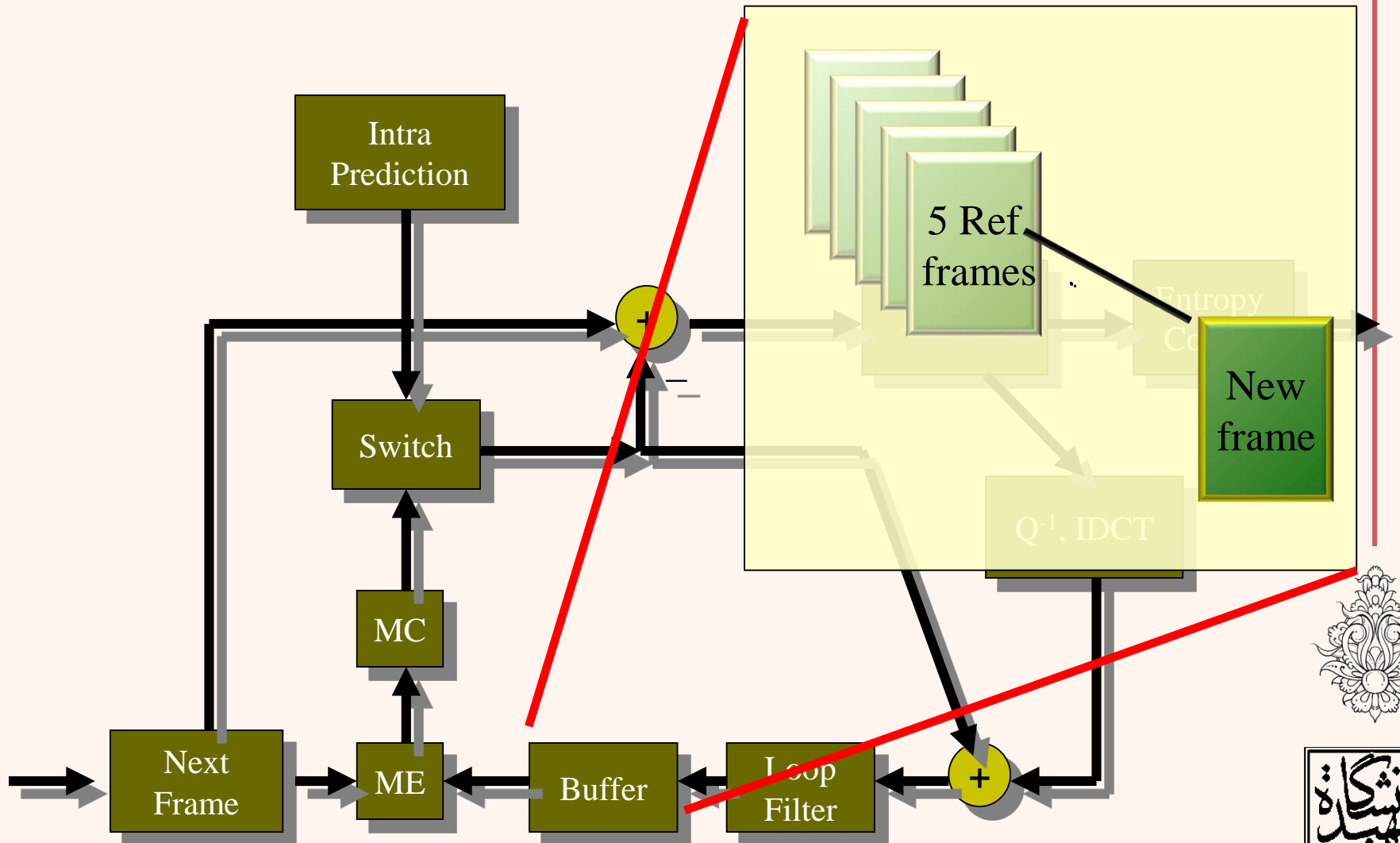
تبدیل صحیح



H.264 جزئیات بهبود صورت گرفته در استاندارد

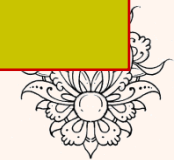
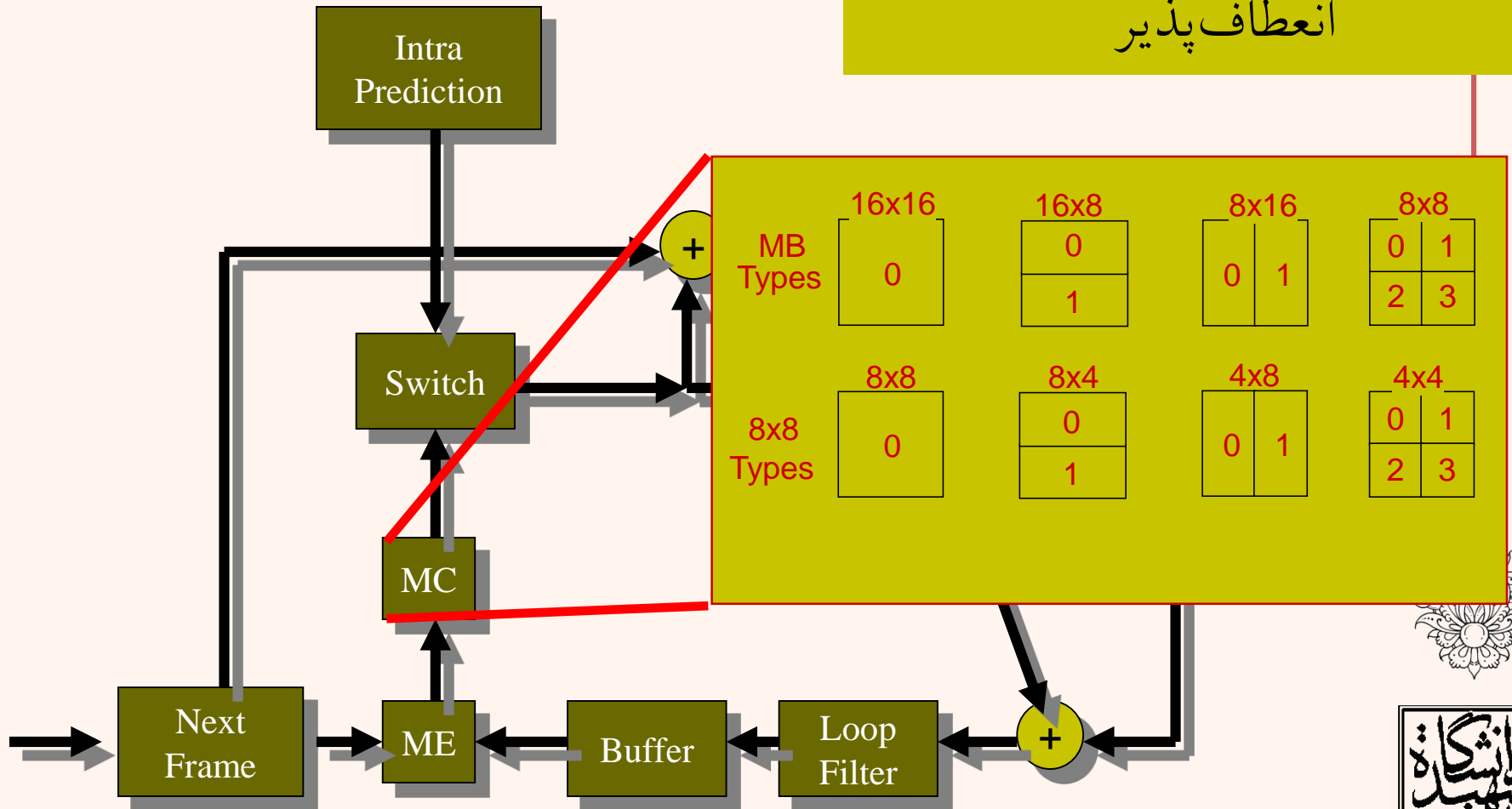
استفاده از Intra Predictions





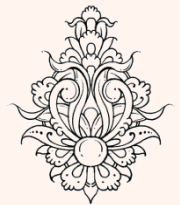
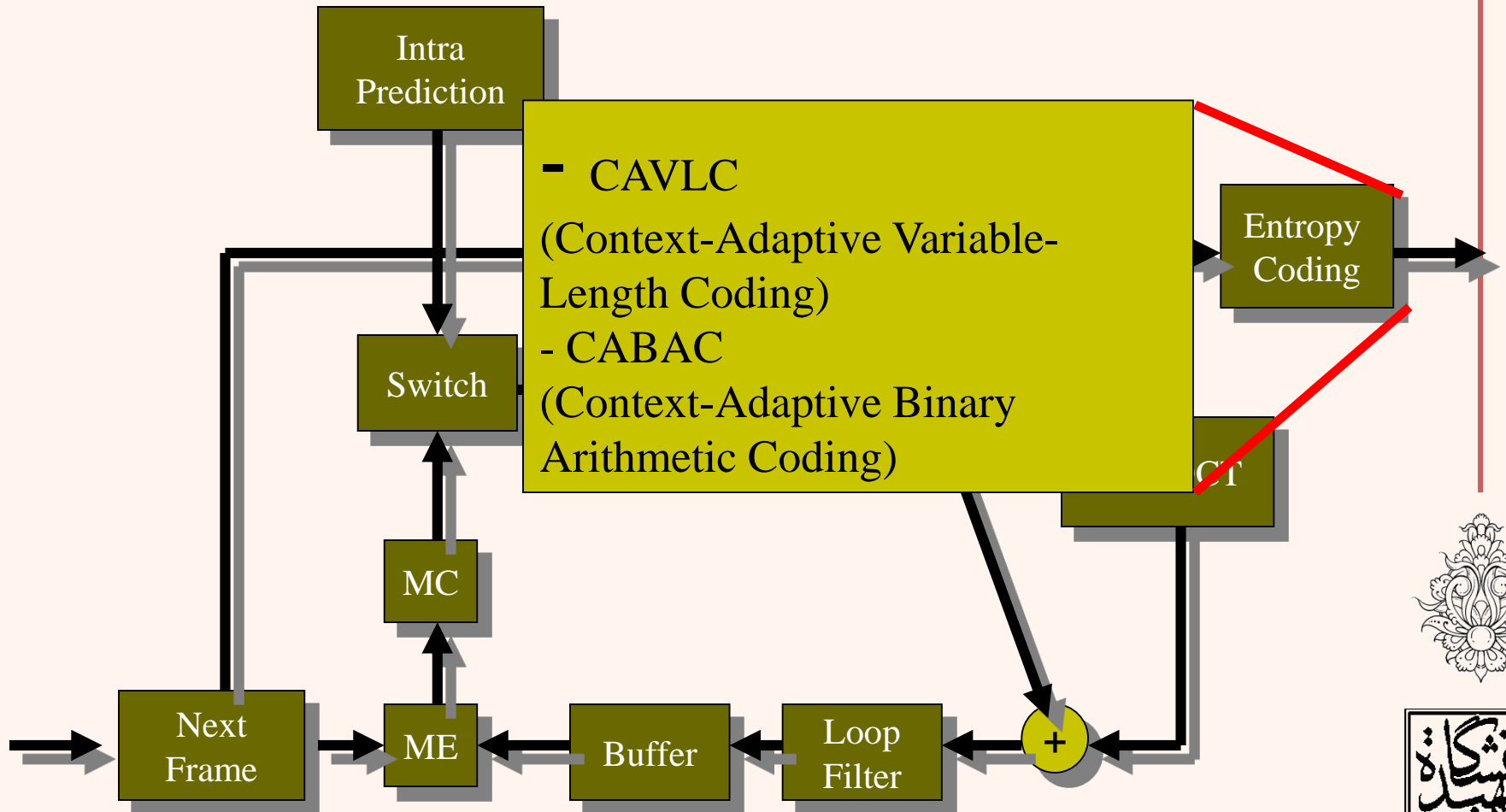
H.264 جزئیات بهبود صورت گرفته در استاندارد

استفاده از ساختار درختی و اندازه‌ی انعطاف‌پذیر



H.264 جزئیات بهبود صورت گرفته در استاندارد

کدینگ آنتروپی



پروفایل‌ها

video-telephony, video-conferencing
and wireless communications

• در H.264 سه پروفایل وجود دارد:

Baseline –

- Intra coding و Inter coding (I Slice/P Slice)
- context-adaptive variable-length codes (CAVLC).

Main –

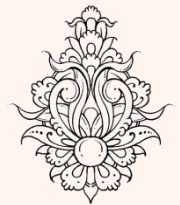
television
broadcasting and video storage

- inter-coding using B-slices
- interlaced video
- context-based arithmetic coding (CABAC)

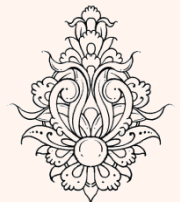
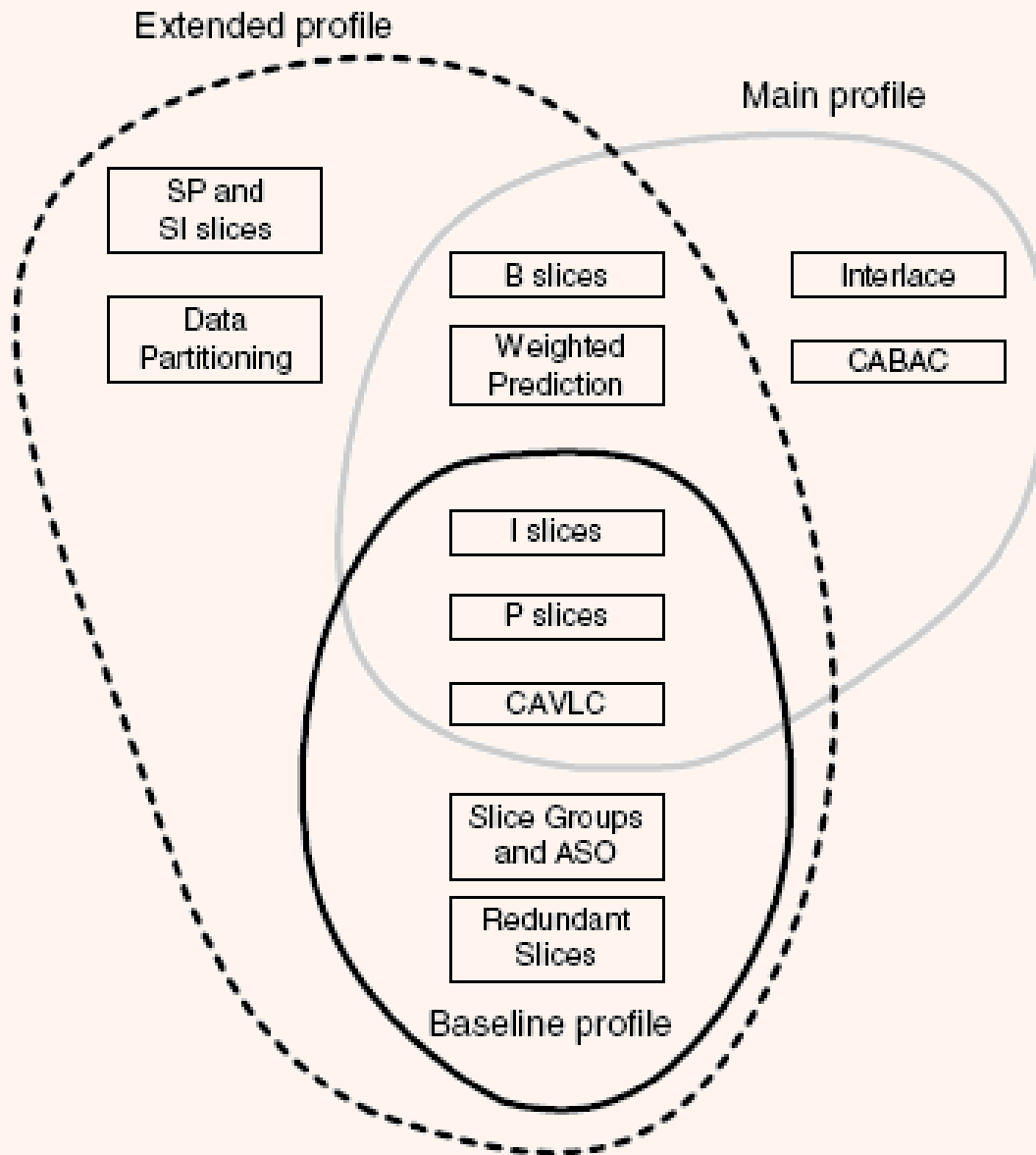
Extended –

- interlaced video را پشتیبانی نمی‌کند.
- CABAC را پشتیبانی نمی‌کند.
- SP- and SI-slices
- improved error resilience

streaming media applications

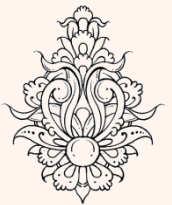


ارتباط میان پروفایل‌ها

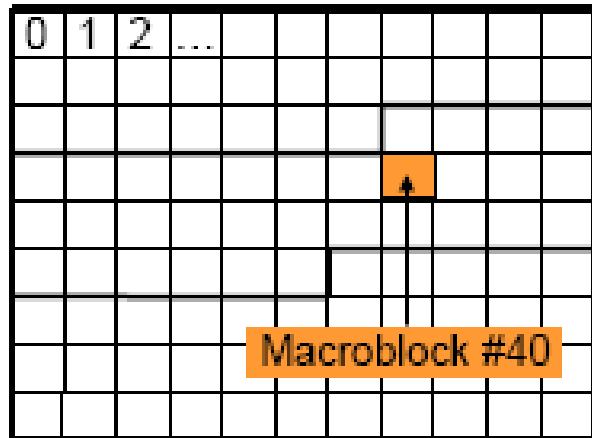
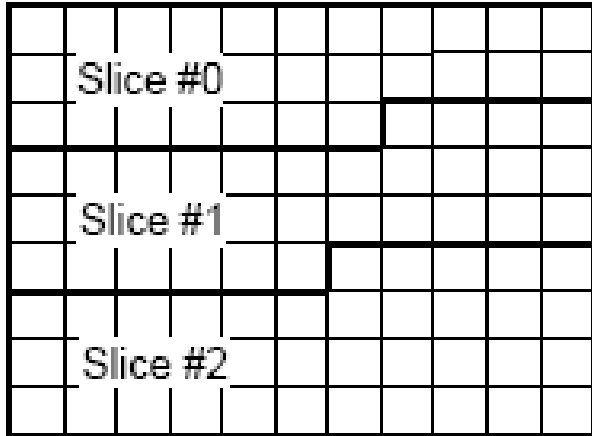


تصاویر مبتنا

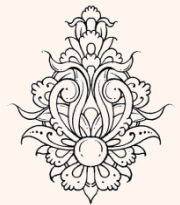
- یک کدگذار H.264 از فریم‌های گذشته‌ی قبلی به عنوان مبتنایی برای تخمین بردارهای جبران حرکت استفاده می‌نماید.
- این مسأله کدگذار را قادر می‌سازد بهترین ماکروبلوک را در میطه‌ی گسترده‌تری انتخاب نماید.
- این انتخاب در استانداردهای قبلی تنها محدود به فریم گذشته‌ی قبلی می‌شد.
- هر تصویر در ساختار ویدئویی به صورت یک یا چند Slice کد می‌گردد.



تصاویر مینا (ادامه...)



- یک تصویر می‌تواند به یک یا چند slice تقسیم بندی شود.
- هر slice شامل گروهی ماکروبلای است که به هم وابسته‌اند.
- هر ماکروبلای بسته به کاربرد می‌تواند مجدد تقسیم بندی شود.



Slice Group

- ماکرو بلاک‌ها معمولاً به صورت Raster Scan جارو می‌شوند.
- می‌توان انواع Slice Group تعریف کرد به گونه‌ای که هر یک برای کاربردی خاص مفید باشد.

Figure 6.6 Slice groups: Interleaved map (QCIF, three slice groups)

0
1
2
0
1
2
0
1
2

Figure 6.7 Slice groups: Dispersed map (QCIF, four slice groups)

				1	2	3	0	1	2	
0	1	2	3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3	0	1	2

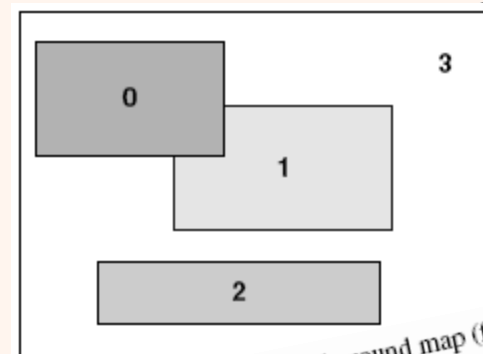
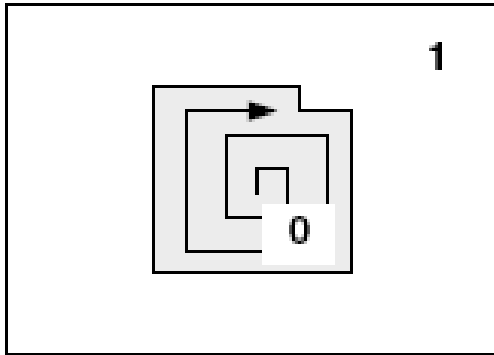


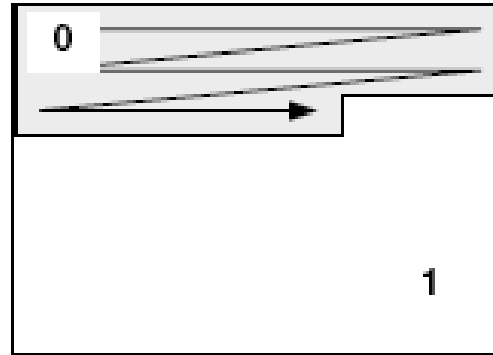
Figure 6.8 Slice groups: Foreground and Background map (four slice groups)

Figure 6.9 Slice groups: Box-out, Raster and Wipe maps

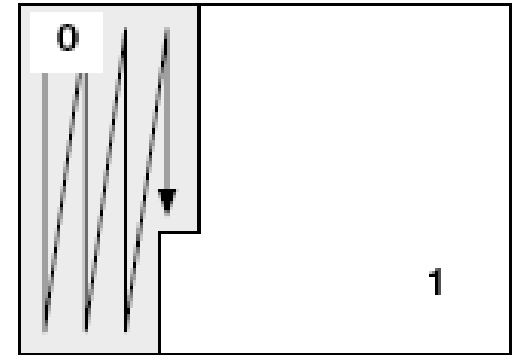
Box-out



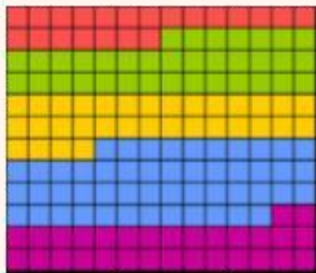
Raster



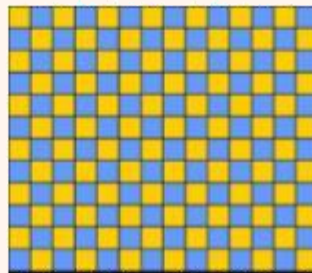
Wipe



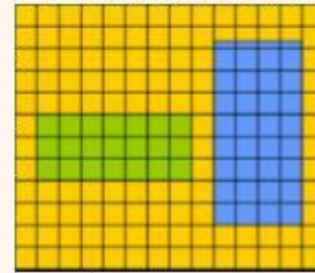
• Type 0



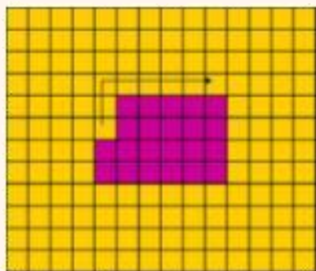
• Type 1



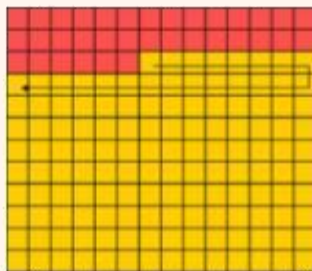
• Type 2



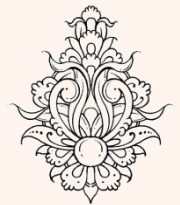
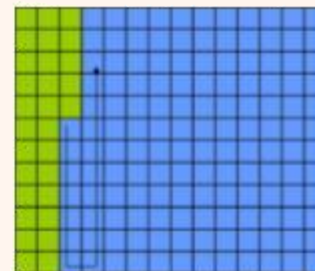
• Type 3

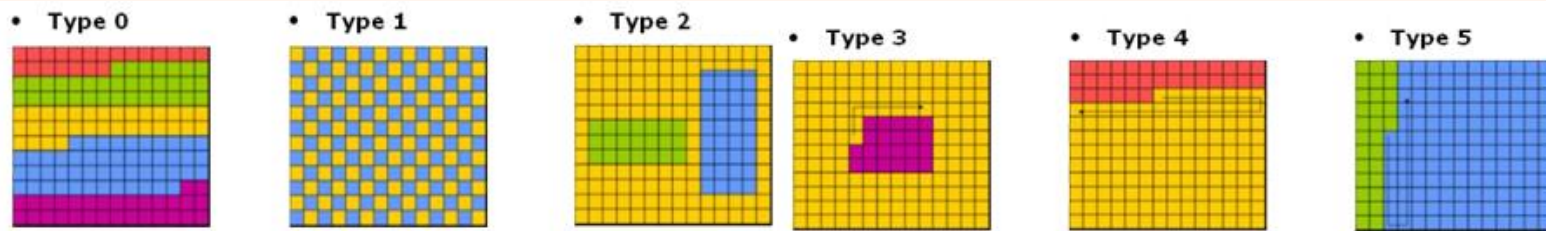


• Type 4



• Type 5





Macroblock to slice group map types

Type	Name	Description
0	Interleaved	run_length MBs are assigned to each slice group in turn (Figure 6.6).
1	Dispersed	MBs in each slice group are dispersed throughout the picture (Figure 6.7).
2	Foreground and background	All but the last slice group are defined as rectangular regions within the picture. The last slice group contains all MBs not contained in any other slice group (the 'background'). In the example in Figure 6.8, group 1 overlaps group 0 and so MBs not already allocated to group 0 are allocated to group 1.
3	Box-out	A 'box' is created starting from the centre of the frame (with the size controlled by encoder parameters) and containing group 0; all other MBs are in group 1 (Figure 6.9).
4	Raster scan	Group 0 contains MBs in raster scan order from the top-left and all other MBs are in group 1 (Figure 6.9).
5	Wipe	Group 0 contains MBs in vertical scan order from the top-left and all other MBs are in group 1 (Figure 6.9).
6	Explicit	A parameter, slice_group_id, is sent for each MB to indicate its slice group (i.e. the macroblock map is entirely user-defined).



انواع Slice

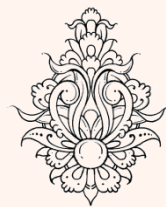
I slice
 I macroblock only
P slice
 P, I macroblock
B slice
 B, I macroblock
SI slice
SP slice

list 0
 past reference frame buffer
list 1
 future reference frame buffer

Slice type	Description	Profile(s)
I (Intra)	Contains only I macroblocks (each block or macroblock is predicted from previously coded data within the same slice).	All
P (Predicted)	Contains P macroblocks (each macroblock or macroblock partition is predicted from one list 0 reference picture) and/or I macroblocks.	All
B (Bi-predictive)	Contains B macroblocks (each macroblock or macroblock partition is predicted from a list 0 and/or a list 1 reference picture) and/or I macroblocks.	Extended and Main
SP (Switching P)	Facilitates switching between coded streams; contains P and/or I macroblocks.	Extended
SI (Switching I)	Facilitates switching between coded streams; contains SI macroblocks (a special type of intra coded macroblock).	Extended

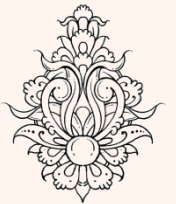
ماکروبلوک

- یک ماکروبلوک یک نامیه به ابعاد 16×16 از یک فریم ویدئویی را شامل می‌شود.
- (بلوک 16×16 روشنایی، بلوک 8×8 برای C_b و بلوک 8×8 برای C_r).
- ماکروبلوک‌ها به صورت پیش‌فرض برای اساس Raster Scan شماره‌گذاری شده، دسترسی به آنها امکان‌پذیر می‌باشد.
- پیش‌بینی ماکروبلوک‌ها بسته به اینکه در چه موقعیتی واقع شده‌اند متفاوت است.



پیش‌بینی ماکروبلوک‌ها

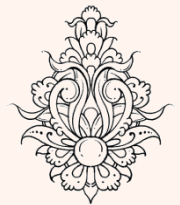
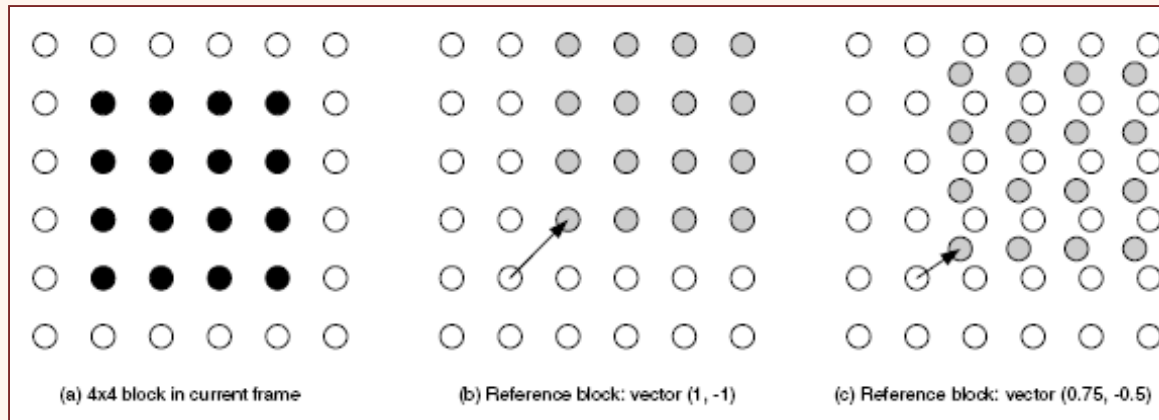
- تمامی پیش‌بینی‌ها و تخمین‌های صورت گرفته، بر اساس داده‌های گذشته‌ی پیشین صورت می‌پذیرد.
 - نمونه‌های کد شده‌ی intra بر اساس نمونه‌های گذشته در slice جاری کدگذاری شده، کدگشایی و دوباره‌سازی می‌گردند.
 - در مقابل نمونه‌هایی که به صورت inter کد می‌شوند بر اساس فریم‌های قبلی که کد شده‌اند کدگذاری می‌شوند.
- پیش‌بینی صورت پذیرفته از ماکروبلوک یا بلوک جاری کسر می‌گردد و میزان اختلاف فشرده گردیده همراه با اطلاعات مورد نیاز جهت تکرار فرایند پیش‌بینی به سمت کدگشا ارسال می‌گردد.
- کدگشا بلوک پیش‌بینی مورد نظر را ساخته و آن را به بلوک باقی مانده کدگشایی شده جهت ساخت داده‌ی اصلی اضافه می‌نماید. بدین‌سان داده‌ی دوباره‌سازی شده در سمت کدگذار و داده‌ی ساخته شده در سمت کدگشا یکسان خواهند بود.



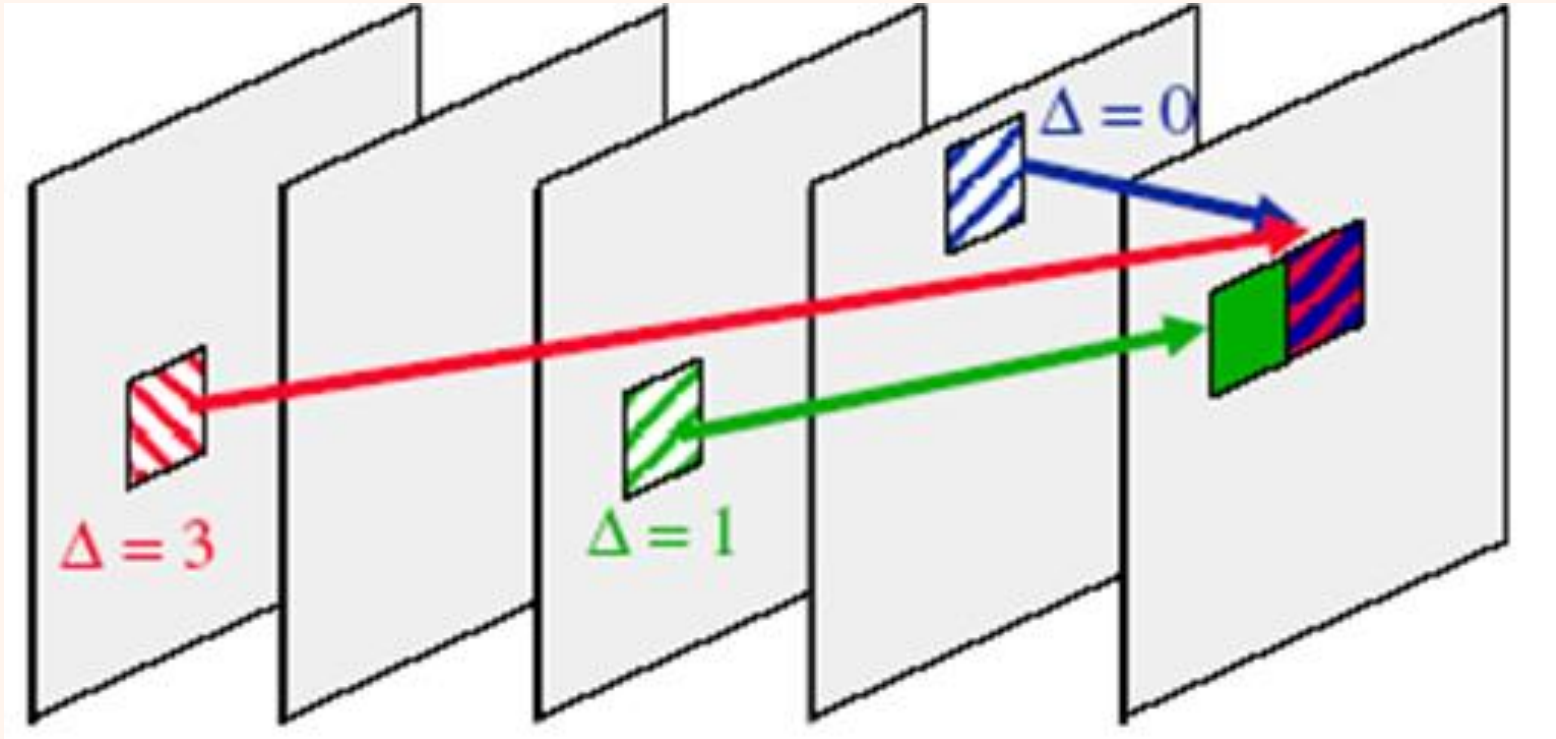
پیش‌بینی inter

• در پیش‌بینی inter، پیش‌بینی بر اساس فریم‌های ویدئو که قبلاً کد شده‌اند به صورت جبران حرکت صورت می‌پذیرد.

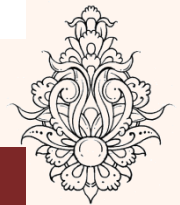
– تفاوت عمده میان استانداردهای پیشین و H.264 پشتیبانی اندازه‌ی بلوک‌های متفاوت از 4×4 تا 16×16 است.



فریم‌های مبتنا

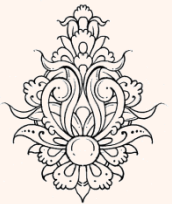


Multiframe motion compensation. In addition to the motion vector, also picture reference parameters (Δ) are transmitted. The concept is also extended to B slices



ادامه...

- مولفه‌ی روشنایی در هر ماکروبلوک (16×16) می‌تواند به چهار گروه شکسته شود.
- جبران حرکت می‌تواند بر اساس یک بلوک 16×16 ، دو بلوک 8×16 یا 16×8 و یا چهار بلوک 8×8 صورت پذیرد.
- اگر مد 8×8 انتخاب گردد، هر بلوک 8×8 می‌تواند به چهار بلوک کوچک‌تر شکسته شود. در این حالت نیز می‌توان یک بلوک 8×8 ، دو بلوک 8×4 یا 4×8 و یا چهار بلوک 4×4 داشته باشیم.



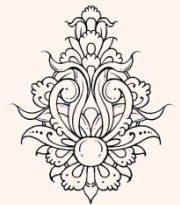
ساختار درختی

- به دلیل ساختار درختی شکستن بلوک‌ها، این شیوه را جبران حرکت بر پایه ساختار درختی نامیده‌اند.
- برای هر زیر ماکروبلوک یک بردار حرکت مستقل نیاز خواهد بود . انتخاب اندازه‌ی بلوک بزرگ‌تر تعداد بیت‌های کم‌تر برای بردارهای حرکت را سبب می‌شود در مقابل پس‌مانده‌های بلوک‌ها انرژی بیشتری را شامل می‌شوند.

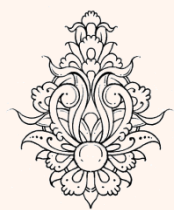
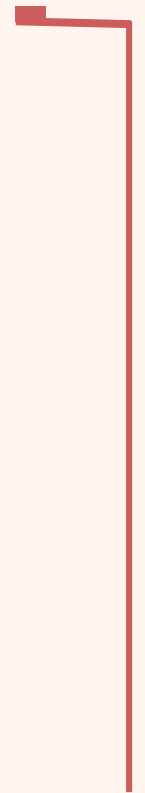
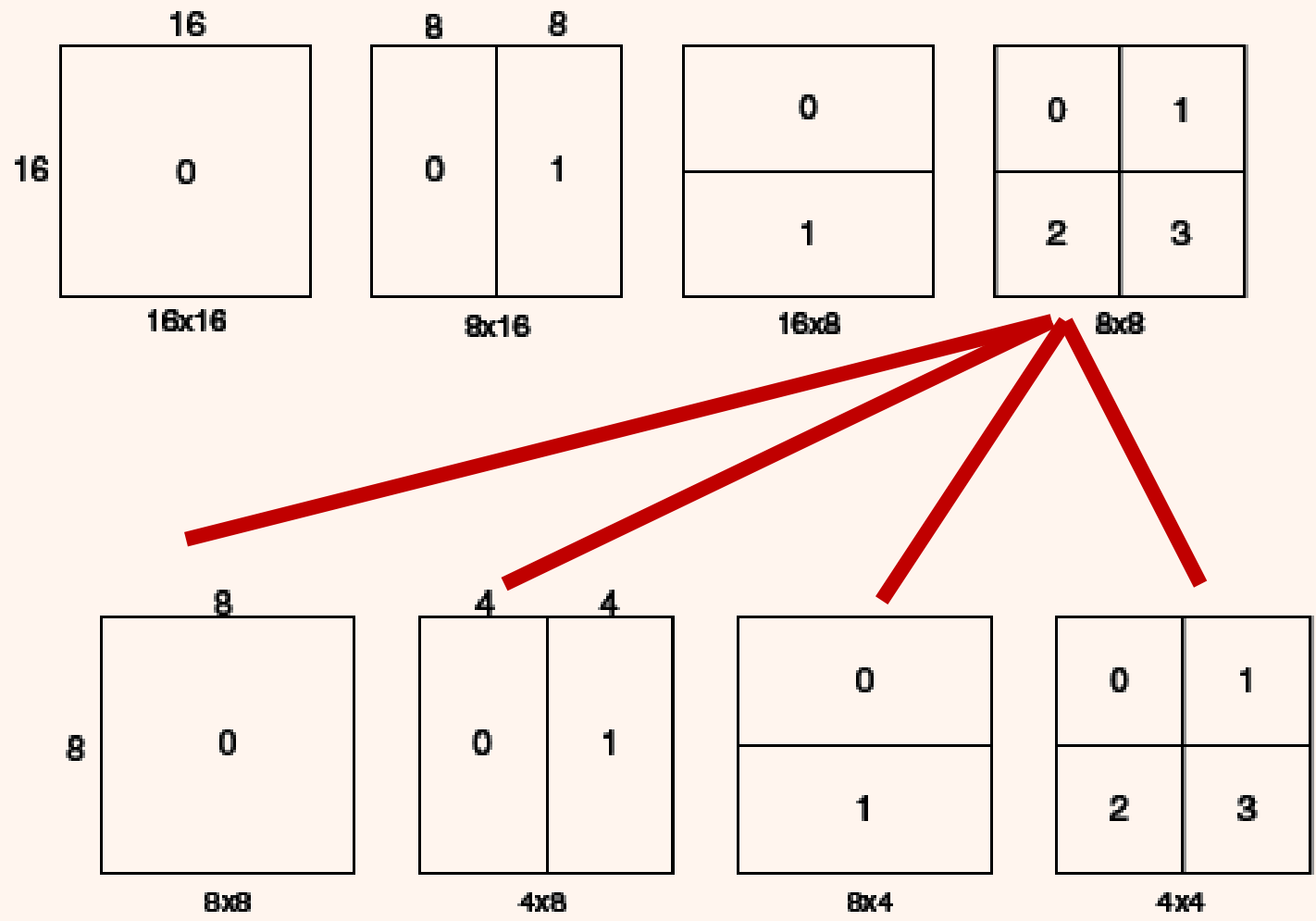


Macroblock partitions

- از منطقی دیگر، وقتی اندازه‌ی بلوک‌ها کوچک‌تر باشد، انرژی کمتری را برای پس‌ماندها سبب می‌شود اما تعداد بیت‌های بیشتری را برای انتخاب بردارهای حرکت سبب خواهد شد.
- بنابراین انتخاب اندازه‌ی بلوک‌ها اثر ویژه‌ای به روی بازده فشرده‌سازی خواهد داشت.
- به صورت کلی انتخاب بلوک بزرگ‌تر برای نواحی هموار و انتخاب بلوک با اندازه‌ی کوچک‌تر برای نواحی با جزئیات بالا مناسب خواهد بود.
- برای C_b و C_r این اندازه‌ها نصف خواهند شد.

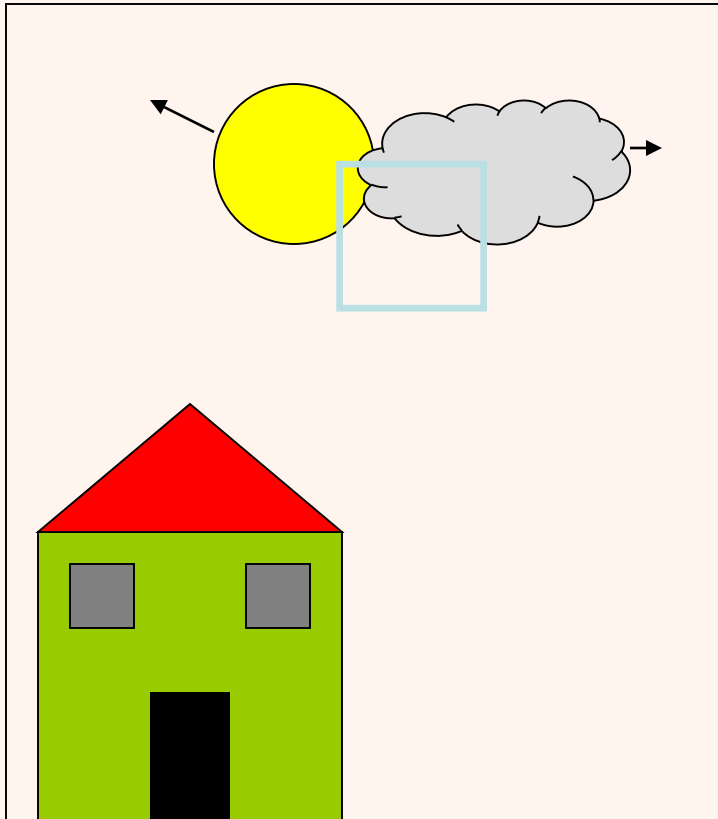


Macroblock partitions

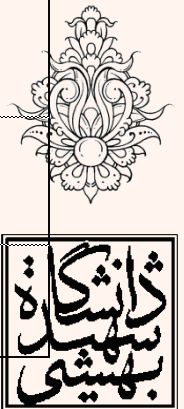
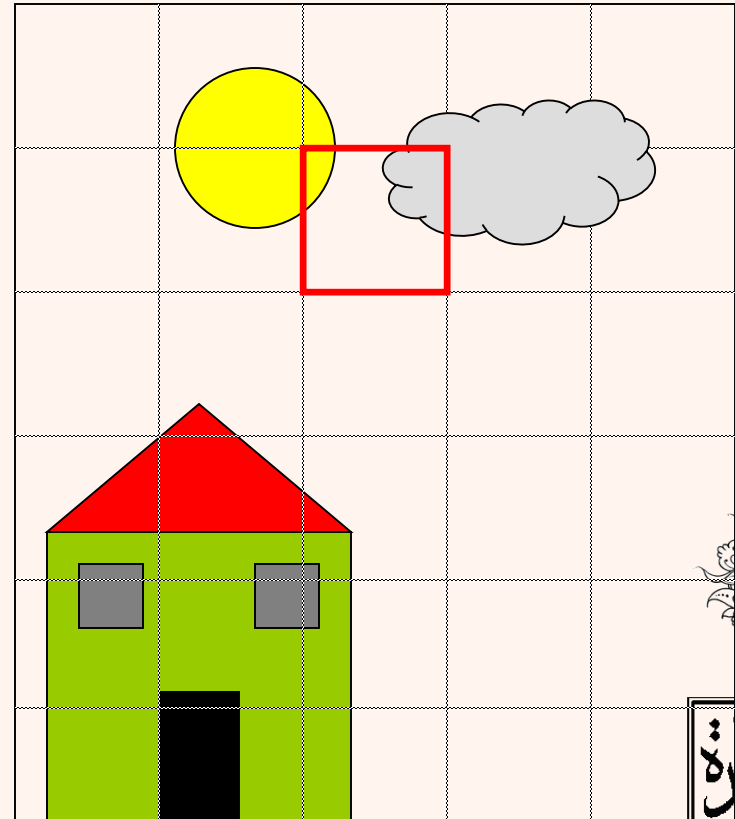


Motion Scale Example

T=1

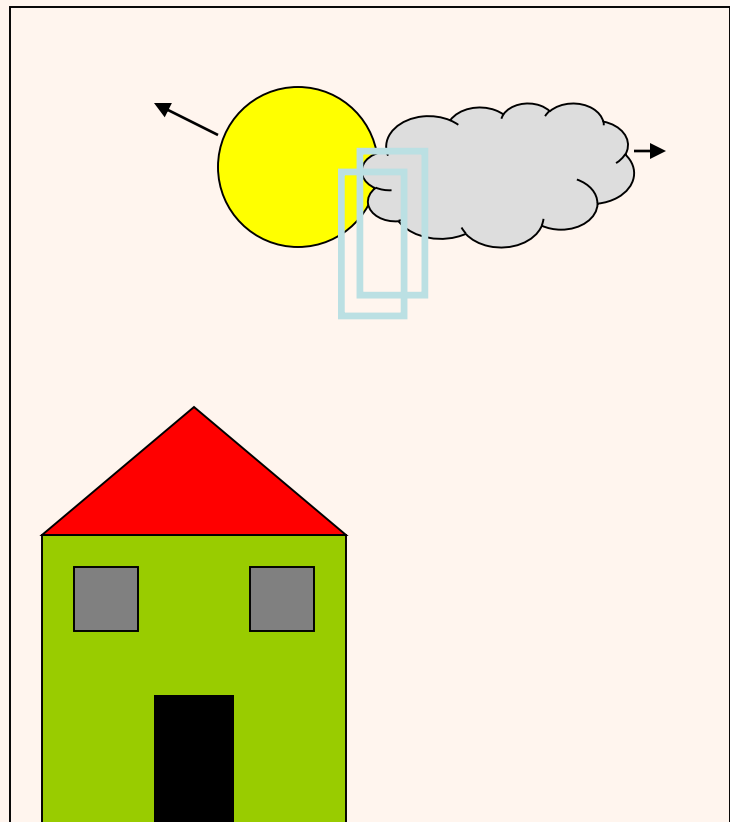


T=2

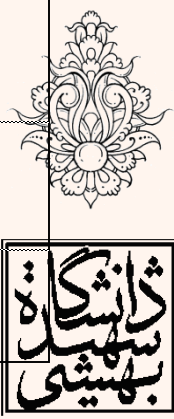
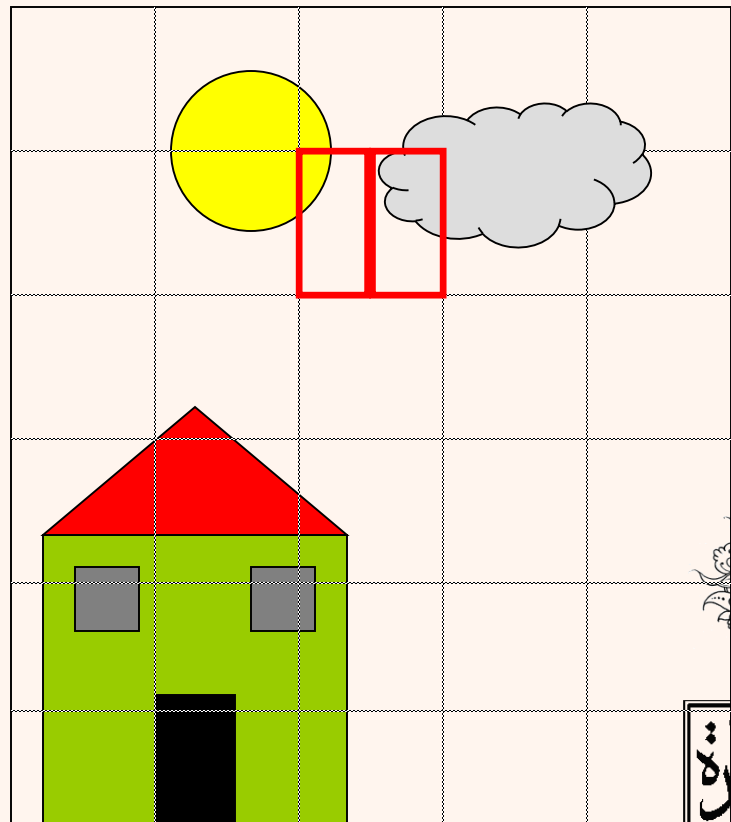


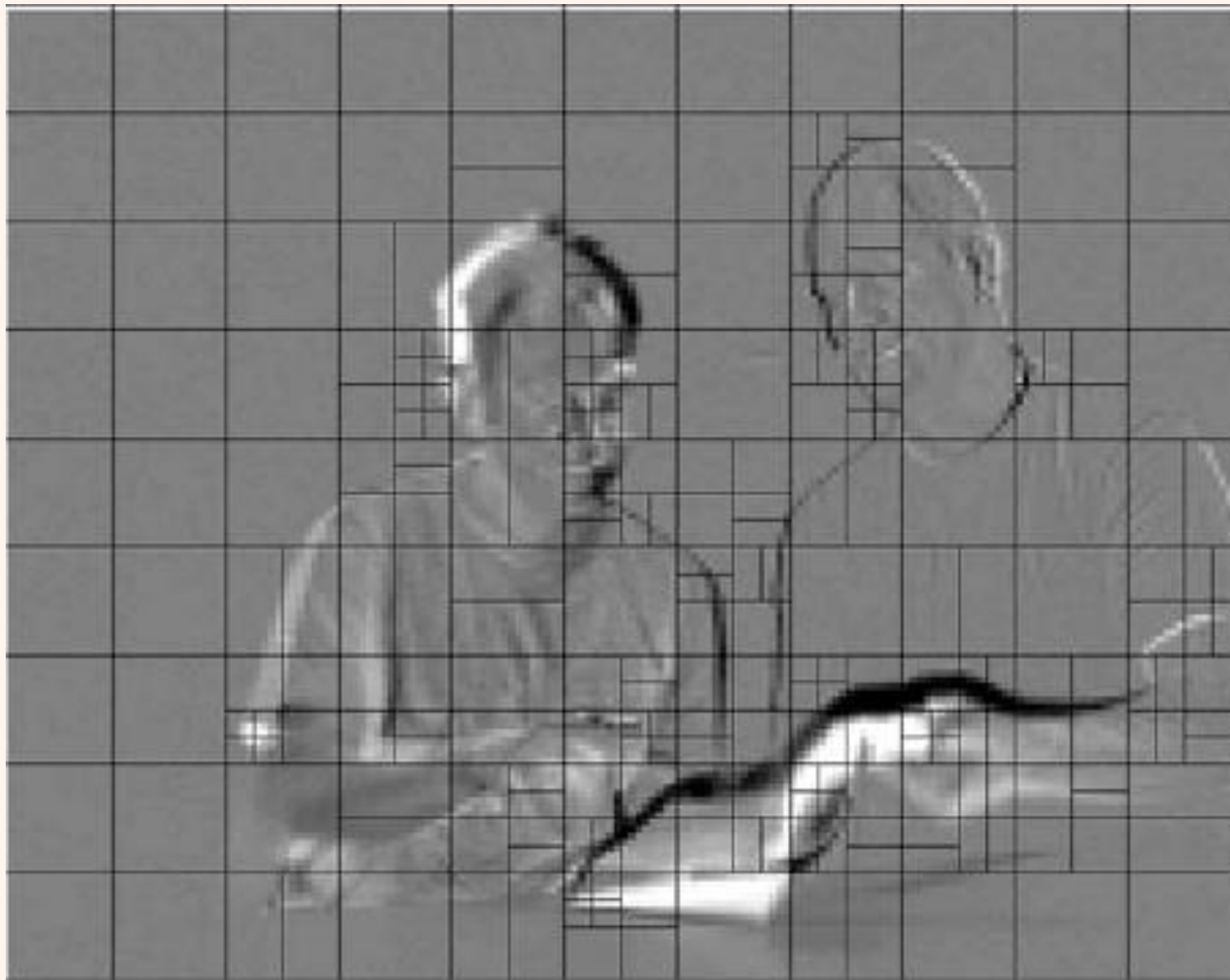
H.264 Variable Block Size Example

T=1



T=2

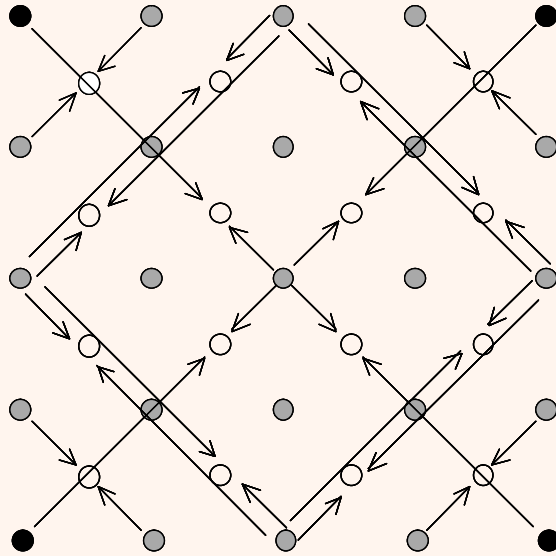




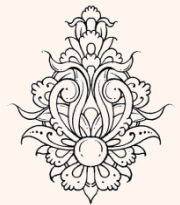
Residual (without MC) showing choice of block sizes

بردارهای حرکت

- بسته به تقسیم‌بندی صورت گرفته مقایسه میان فریم مرجع و فریم جاری برای اندازه بلوک مورد نظر صورت گرفته، تغییر مکان به عنوان بردار حرکت کد می‌شود.



- Integer position pixels
- 1/8 pixels
- 1/2 and 1/4 pixels



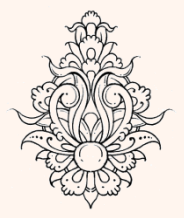
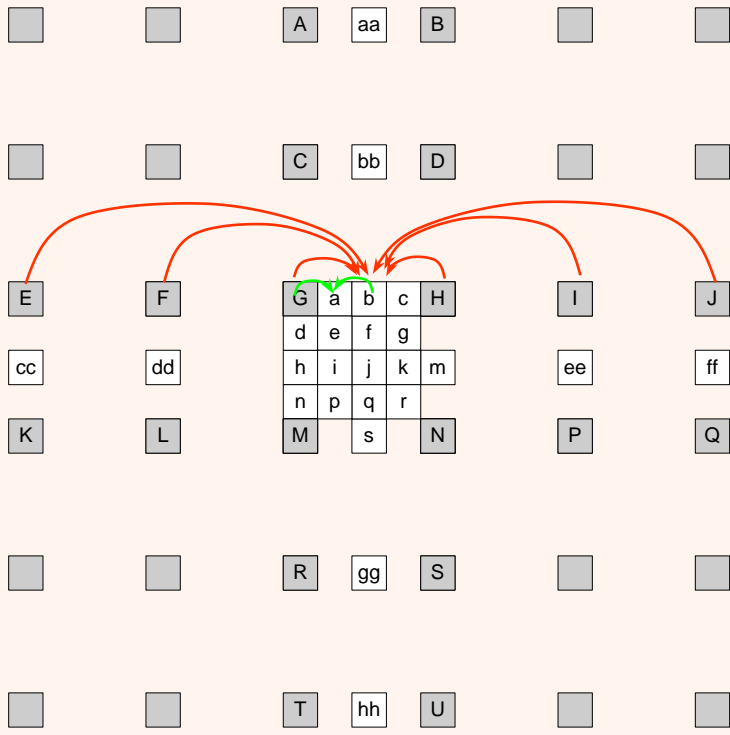
Kwon, S.-k., A. Tamhankar, et al. (2006). "Overview of H.264/MPEG-4 part 10." *Journal of Visual Communication and Image Representation* 17(2): 186-216

• دقت 1/2 پیکسل

$$b = \text{round}((E-5F+20G+20H-5I+J)/32)$$

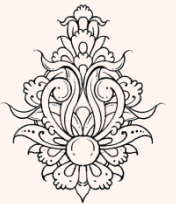
• دقت 1/4 پیکسل

$$a = \text{round}((G+b)/2)$$



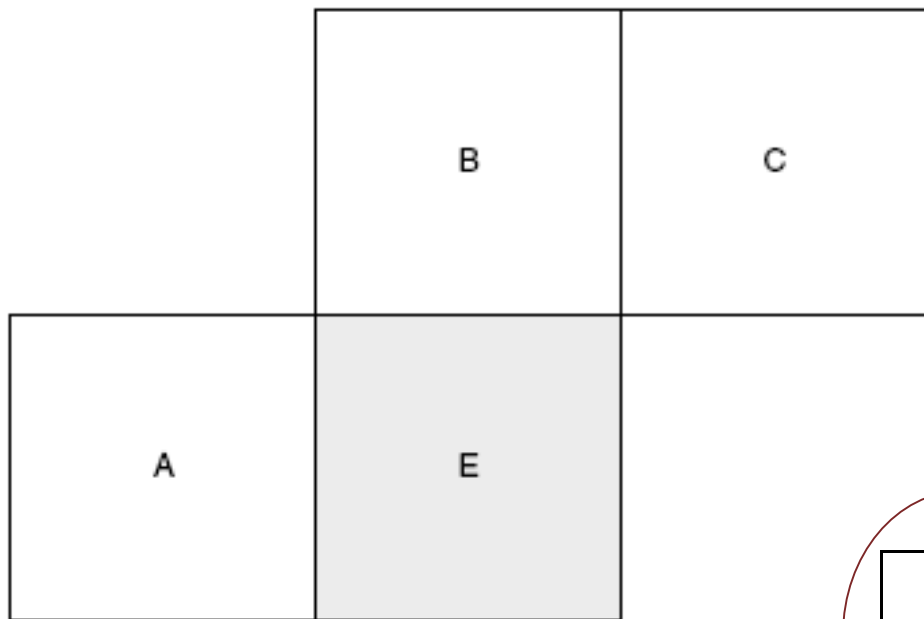
پیش‌بینی بردار حرکت

- بردارهای حرکت که در همسایگی هم هستند بسیار شبیه خواهند بود.
- بدین منظور برای پیش‌بینی بردار حرکت یا همان MV_p از اطلاعات بردارهای همسایه استفاده می‌کنیم.
- چیزی که ارسال و کد می‌گردد MV_D نام دارد که همان اختلاف بردار حرکت جاری و پیش‌بینی شده است.
- در پیش‌بینی اندازه‌ی قسمت‌ها و در دسترس بودن بردارهای همسایه اهمیت دارد.



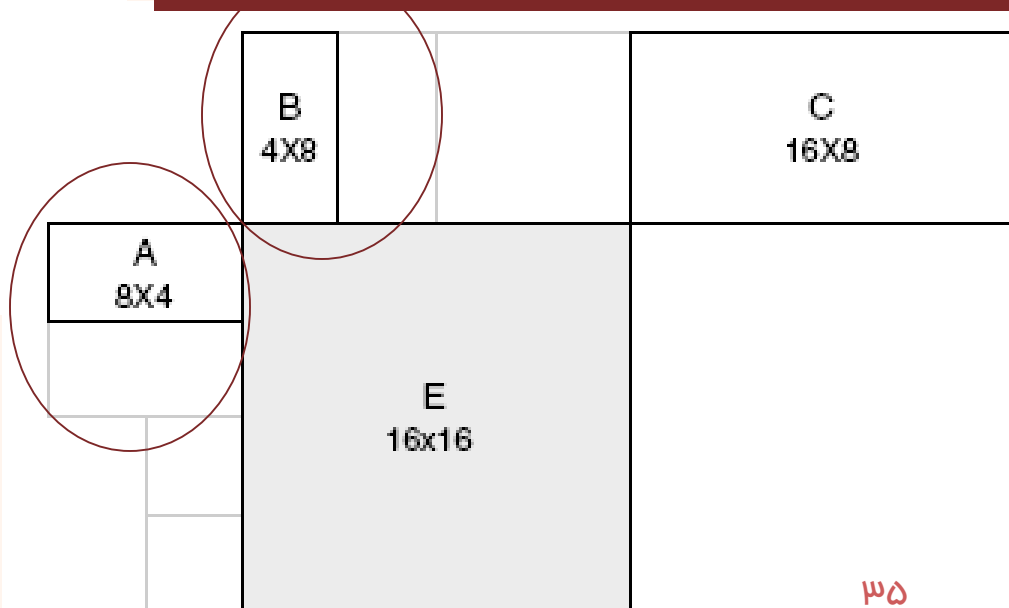
پیش‌بینی بردار حرکت

Motion vectors for neighboring partitions are often highly correlated and so each motion vector is predicted from vectors of nearby, previously coded partitions.



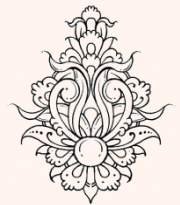
Current and neighboring partitions
(same partition sizes)

Current and neighboring partitions
(different partition sizes)



پیش‌بینی intra

- بر خلاف استانداردهای پیشین که در آن برای فریم‌های از نوع I هیچ‌گونه پیش‌بینی صورت نمی‌پذیرفت، در استاندارد H.264 برای این دست فریم‌ها پیش‌بینی به وسیله‌ی استفاده از بلوک‌های قبلی فریم جاری انجام می‌شود.
- قبل از کدگذاری بلوک مورد نظر مقدار پیش‌بینی شده از بلوک جاری کسر شده و بلوک پس‌ماند را برای ارسال تشکیل می‌دهد.
- برای Luma یا مؤلفه‌ی روشنایی مقادیر پیش‌بینی برای هر بلوک 4×4 و یا هر بلوک 16×16 تشکیل می‌شود.



پیش‌بینی intra

• برای نمونه‌های luma

• Intra 4×4

– نه اسلوب پیش‌گویی

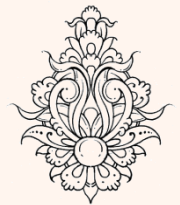
• Intra 16×16

– چهار اسلوب پیش‌گویی

• I_PCM

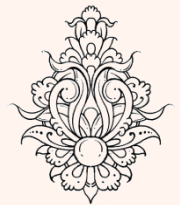
– مقادیر به دست آمده بدون پیش‌گویی

ارسال می‌شود.



پیش‌بینی intra (ادامه...)

- در 4×4 Intra یا $I4$ هر بلوک 4×4 به صورت جداگانه کد شده و برای نوامی با جزییات بالا مناسب است.
- در مقابل 16×16 Intra یا $I16$ برای کد نمودن نوامی هموار مناسب می‌باشد.
- تعداد نه اسلوب برای هر بلوک روشنایی 4×4
- چهار اسلوب برای بلوک روشنایی 16×16
- چهار اسلوب برای مولفه‌های رنگی وجود دارد.
- کدگذار اسلوبی را به عنوان اسلوب بهینه انتخاب می‌نماید که تفاوت میان مقدار پیش‌بینی و بلوک جاری را کمینه نماید.



اسلوب بهينه

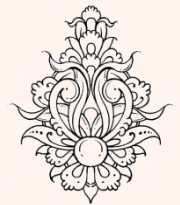
- كدگذار اسلوبى را به عنوان اسلوب بهينه انتخاب مى نمايد كه تفاوت ميان مقدار پيش بينى و بلوك جارى را كمينه نمايد.

Rate-Distortion Optimization

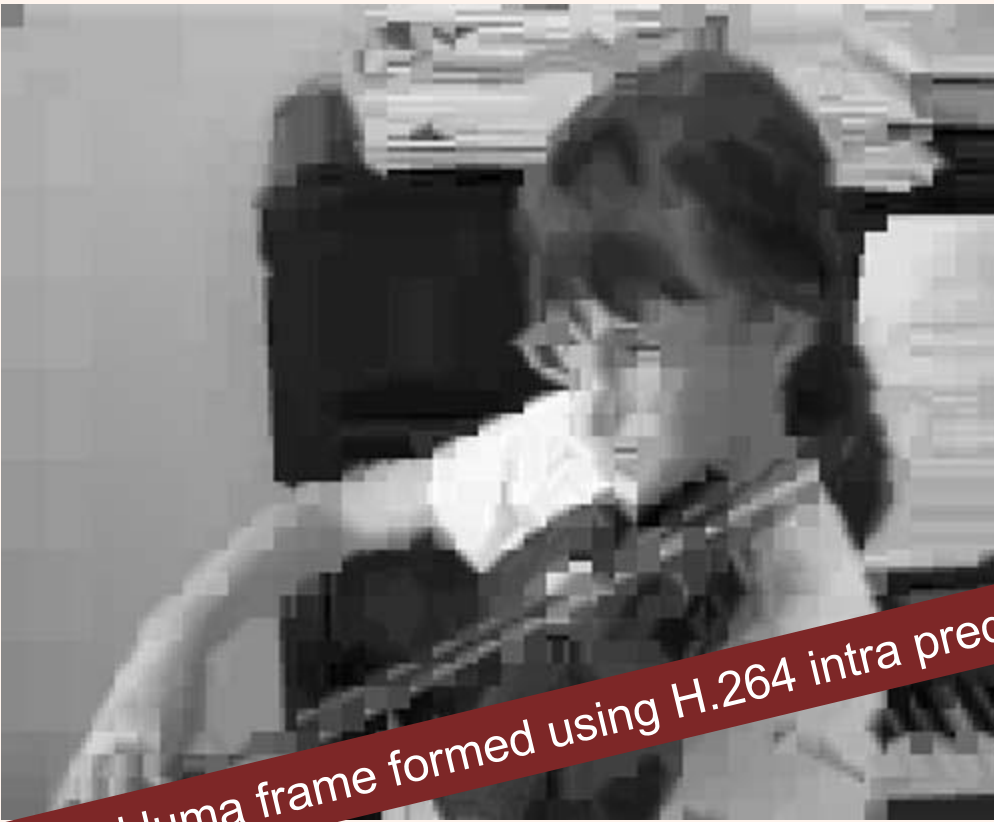
- The goal of RDO is to minimize the perceived distortion with the number of encoded bits subject to a rate constraint

$$\min_{\Omega} J(\Omega, \lambda) = D(\Omega) + \lambda R(\Omega),$$

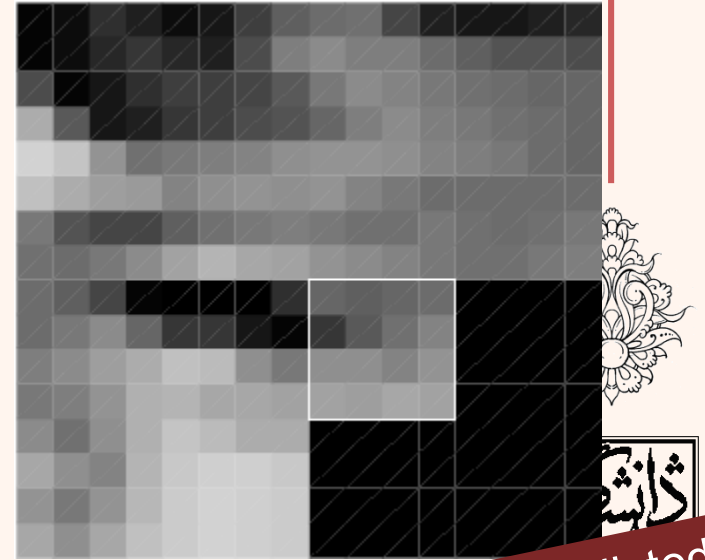
where Ω represents the set of encoder decisions for the block, $D(\Omega)$ and $R(\Omega)$ are the distortion and rate measurement using Ω respectively. R_c is the rate constraint.



مثال



predicted luma frame formed using H.264 intra prediction



4 × 4 luma block to be predicted

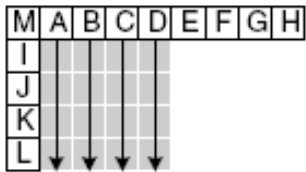
پیش‌بینی intra (۴×۴)

نواحی با جزئیات بالا

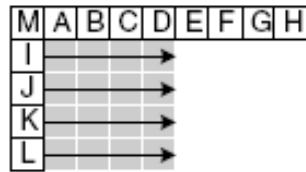
M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Labeling of prediction samples (4 × 4)

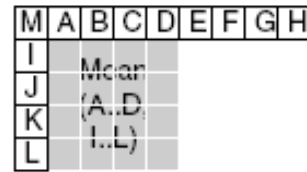
0 (vertical)



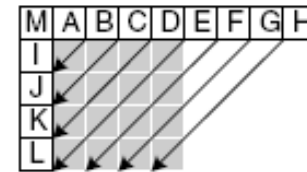
1 (horizontal)



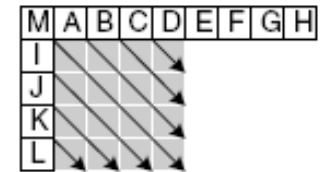
2 (DC)



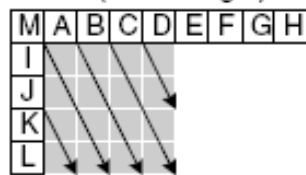
3 (diagonal down-left)



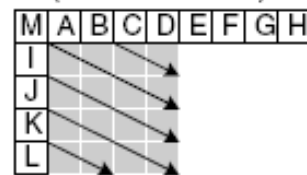
4 (diagonal down-right)



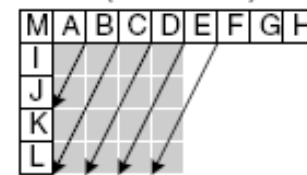
5 (vertical-right)



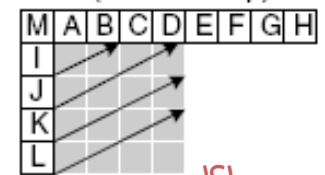
6 (horizontal-down)



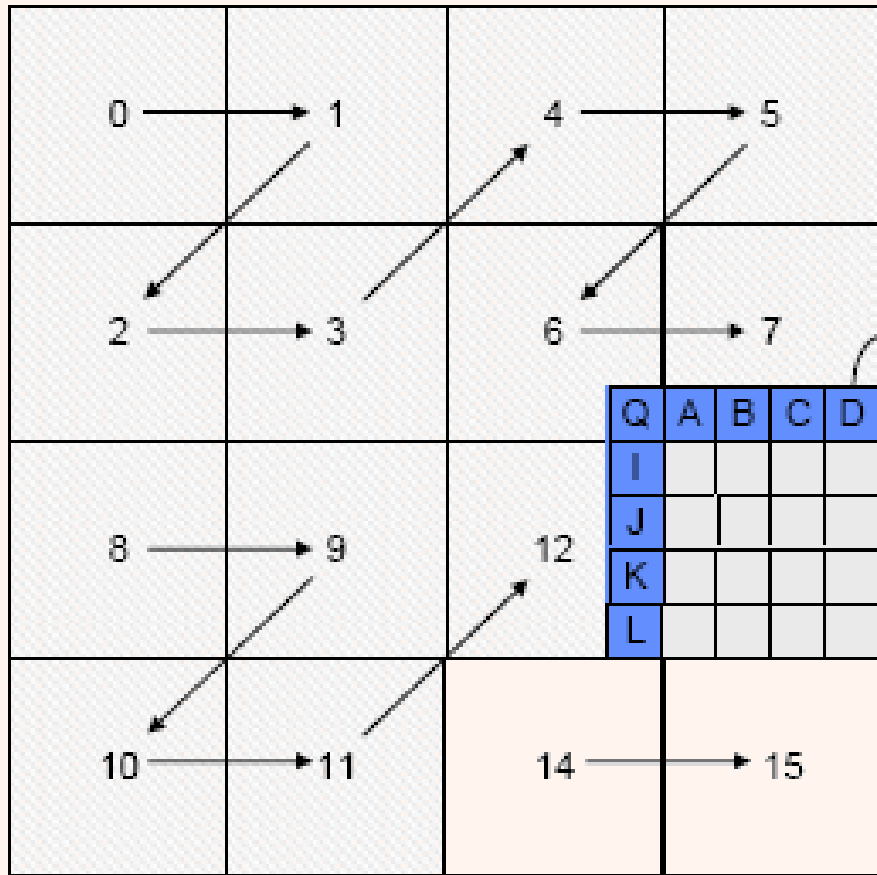
7 (vertical-left)



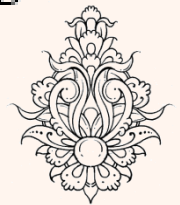
8 (horizontal-up)



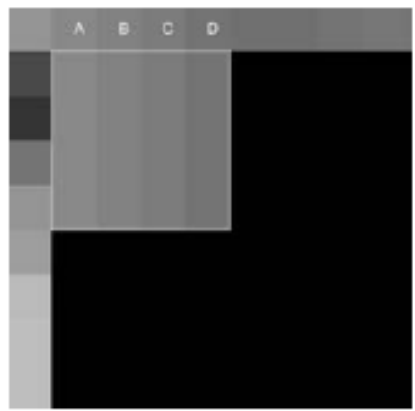
4x4 Boundary Conditions



EFGH not available since this 4x4 block is outside the macroblock – replace EFGH with value of D



مثال



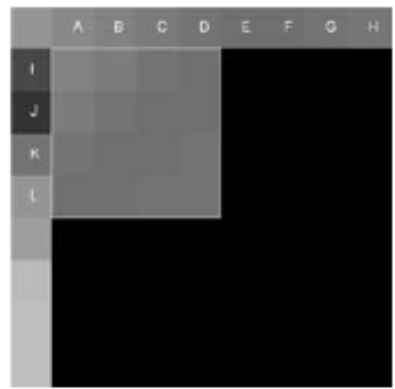
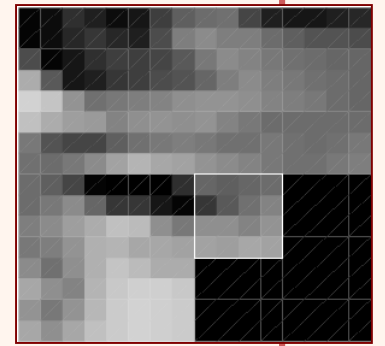
0 (vertical), SAE = 317



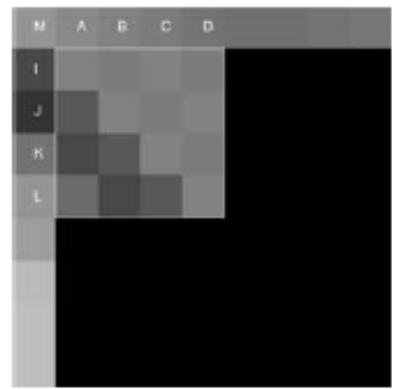
1 (horizontal), SAE = 401



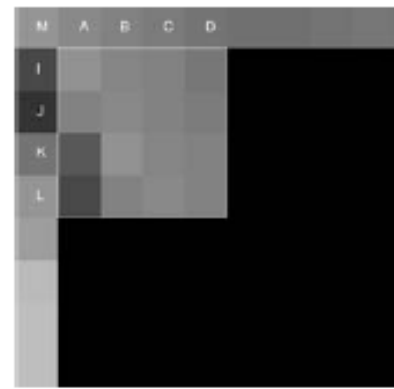
2 (DC), SAE = 317



3 (diag down/left), SAE = 350



4 (diag down/right), SAE = 466



5 (vertical/right), SAE = 419



6 (horizontal/down), SAE = 530



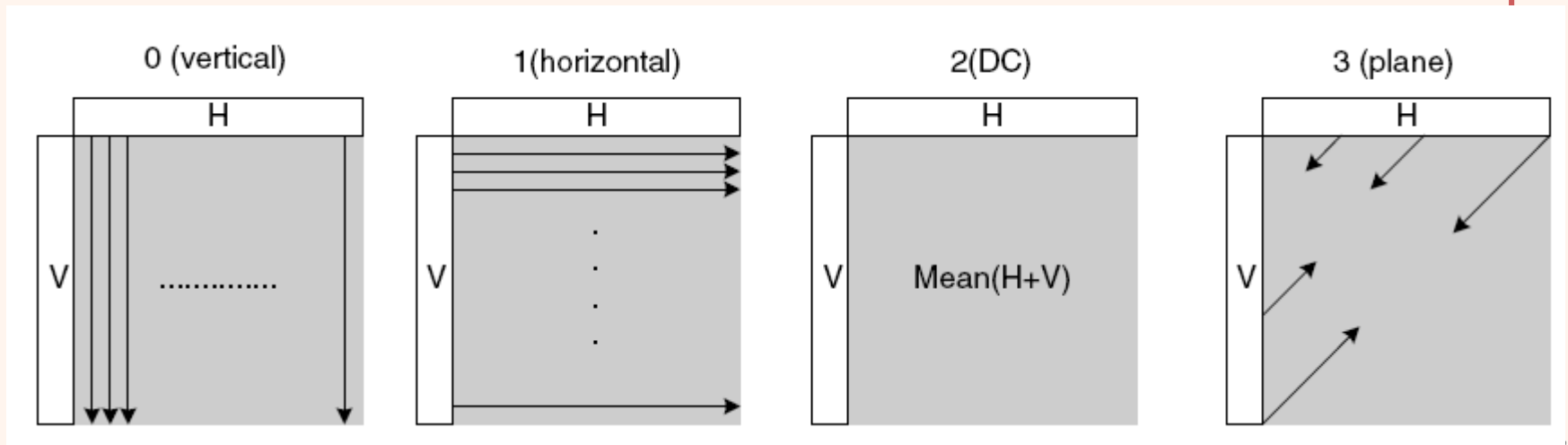
7 (vertical/left), SAE = 351



8 (horizontal/up), SAE = 203

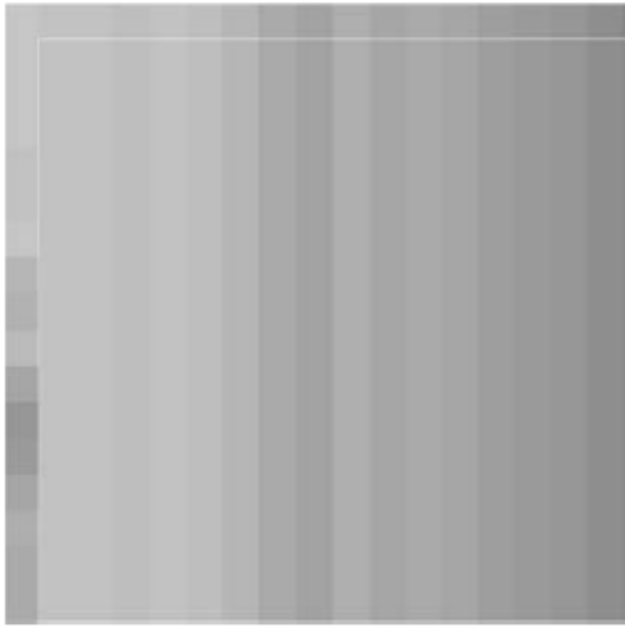
پیش‌بینی (۱۶×۱۶) intra

نواحی هموار



Intra 16 × 16 prediction modes





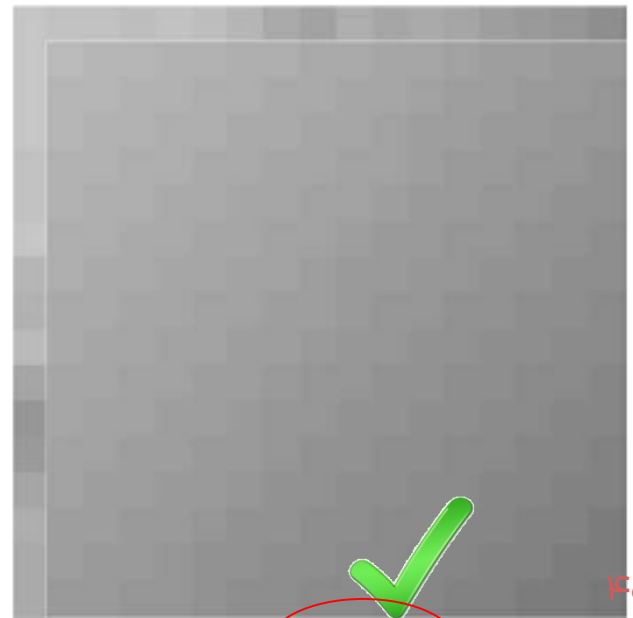
0 (vertical), SAE = 3985



1 (horizontal), SAE = 5097



2 (DC), SAE = 4991



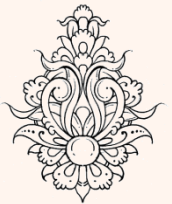
3 (plane), SAE = 2539

$$\text{RD-Cost}(\text{mode}) = D(\text{mode}) + \lambda \times R(\text{mode})$$

- **RD-Cost(mode)** is the rate-distortion cost for a particular mode for a block, and **D(mode)** represents the distortion if the block is coded with that mode, and **R(mode)** is the bit-rate produced if the block is coded with that particular mode.

For calculating each cost the encoder needs to **transform, quantize and entropy code** a block with all the modes, to calculate **R(mode)** because **R(mode)** is the amount of bits used to encode a block with that mode.

Also encoder has to perform entropy **decoding, dequantization and inverse transform** to reconstruct the image on the encoder side itself, to calculate the **D(mode)** because **D(mode)** is the difference between the **original** image and the image after **reconstruction**.



ارسال اسلوب‌های intra به کدگشا

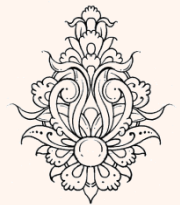
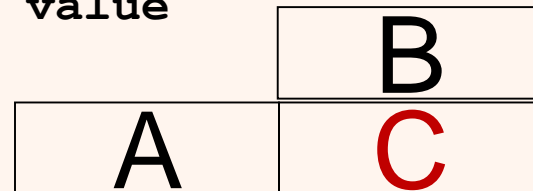
- اسلوب انتخاب‌شده می‌باید به کدگشا ارسال گردد.
- اسلوب‌های همسایه برای بلوک‌های 4×4 بسیار شبیه‌هم هستند.

– از پیش‌گویی برای ارسال این داده‌ها نیز استفاده شود.

- برای ارسال اسلوب‌های intra برای بلوک‌های 16×16 و بلوک‌های رنگی خود مقادیر ارسال می‌شود.

If A and B are available, $C = \min (A, B)$

else If either of these neighboring blocks is not available, the corresponding value A or B is set to 2 (DC).



rem_intra4x4_pred_mode	prediction mode for block
0	0
1	2
2	3
3	4
4	5
5	6
6	7
7	8

Mode=1



Mode=3

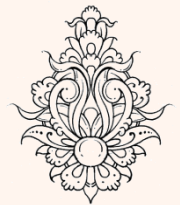
Choice of prediction mode (most probable mode = 1)

- Flag `prev_intra4x4_pred_mode`
 - 1: most probable prediction mode is used.
 - 0: Flag `rem_intra4x4_pred_mode` indicates the change of the mode

صرفه جویی در یک بیت به وسیله کد نمودن با سه بیت

- If `rem_intra4x4_pred_mode < most_probable_mode`,
the prediction mode = rem intra4x4 pred mode
- else

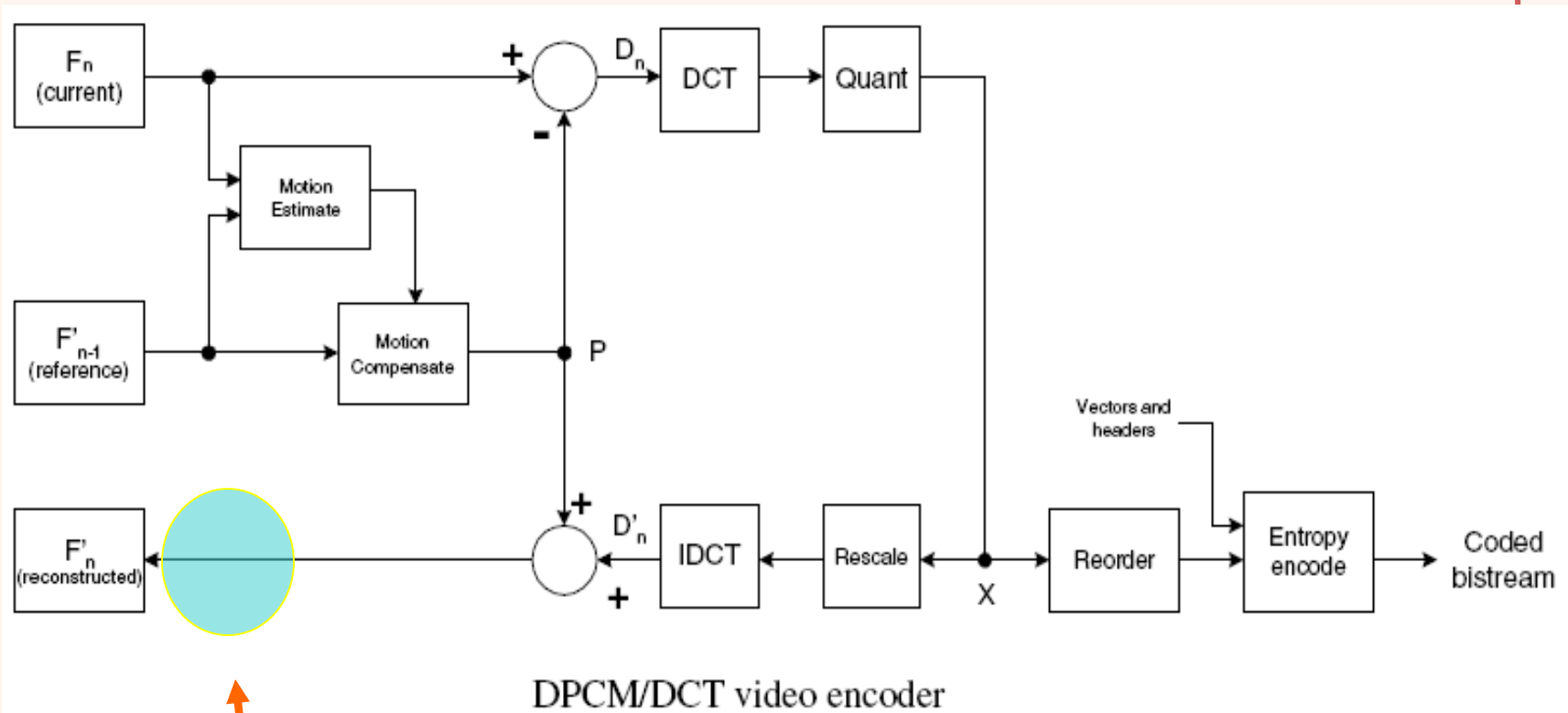
the prediction mode = $(\text{rem_intra4x4_pred_mode} + 1)$



همواره می توان **mpm** را مناسبه نمود

برول به دو تکه شکسته می شود یا **mpm** ارسال می شود یا با **rem** و با استفاده از سه بیت ارسال صورت می گیرد
فشرده سازی

Deblocking Filter

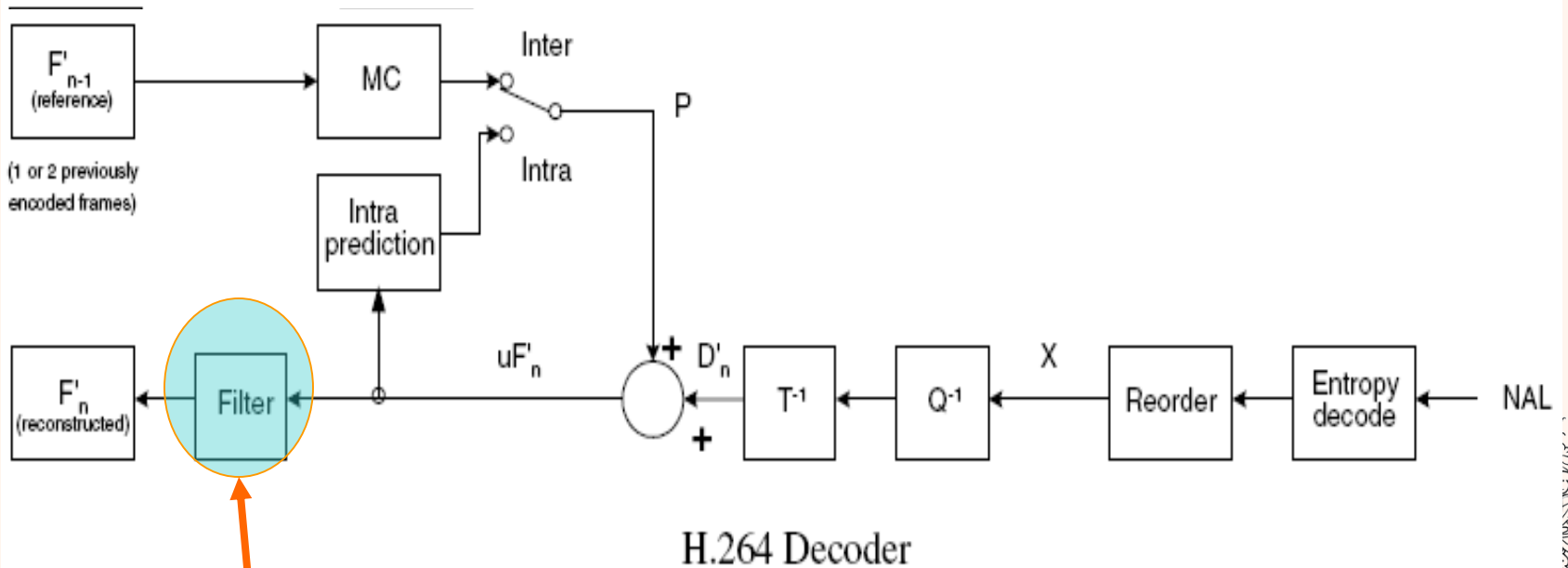


Deblocking Filter



Deblocking Filter

The filtered image is used for motion-compensated prediction of future frames and this can improve compression performance because the filtered image is often a more faithful reproduction of the original frame than a blocky, unfiltered image



Deblocking
Filter



مثال ۱



Non Deblocked Image



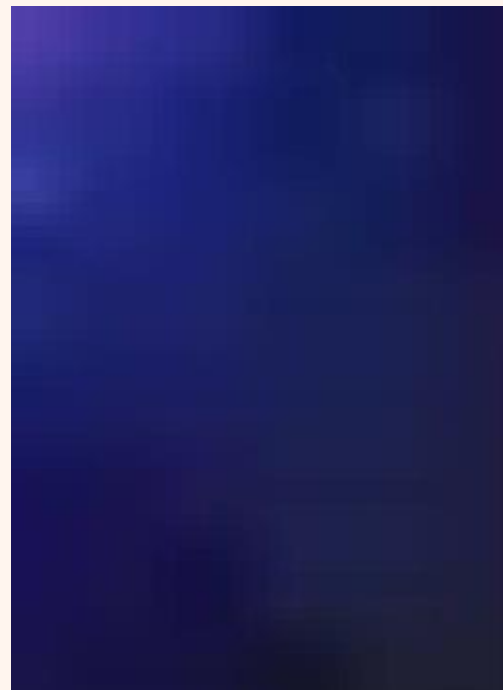
Deblocked Image



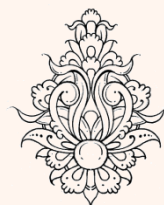
مثال ۲



Non Deblocked Image



Deblocked Image



مثال ۳



Non Deblocked Image

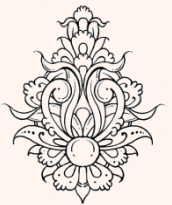


Deblocked Image



ارسال پس‌ماند

- همانند استانداردهای پیشین، H.264 نیز از تبدیل داده‌های پس‌ماند حاصل از پیش‌بینی جهت کدگذاری جریان داده استفاده می‌نماید.
- این مساله در استانداردهای پیشین چون MPEG.1 و MPEG.2 به وسیله‌ی اعمال تبدیل گسسته‌ی کسینوسی به روی بلوک‌های 8×8 صورت می‌گیرد.
- این در حالی است که H.264 بر حسب نوع داده‌ی پسماند که می‌باید کد شود، سه نوع تبدیل را به کار می‌گیرد.



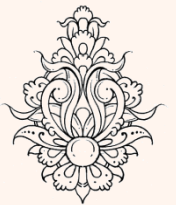
تبدیل‌ها

• سه گونه تبدیل

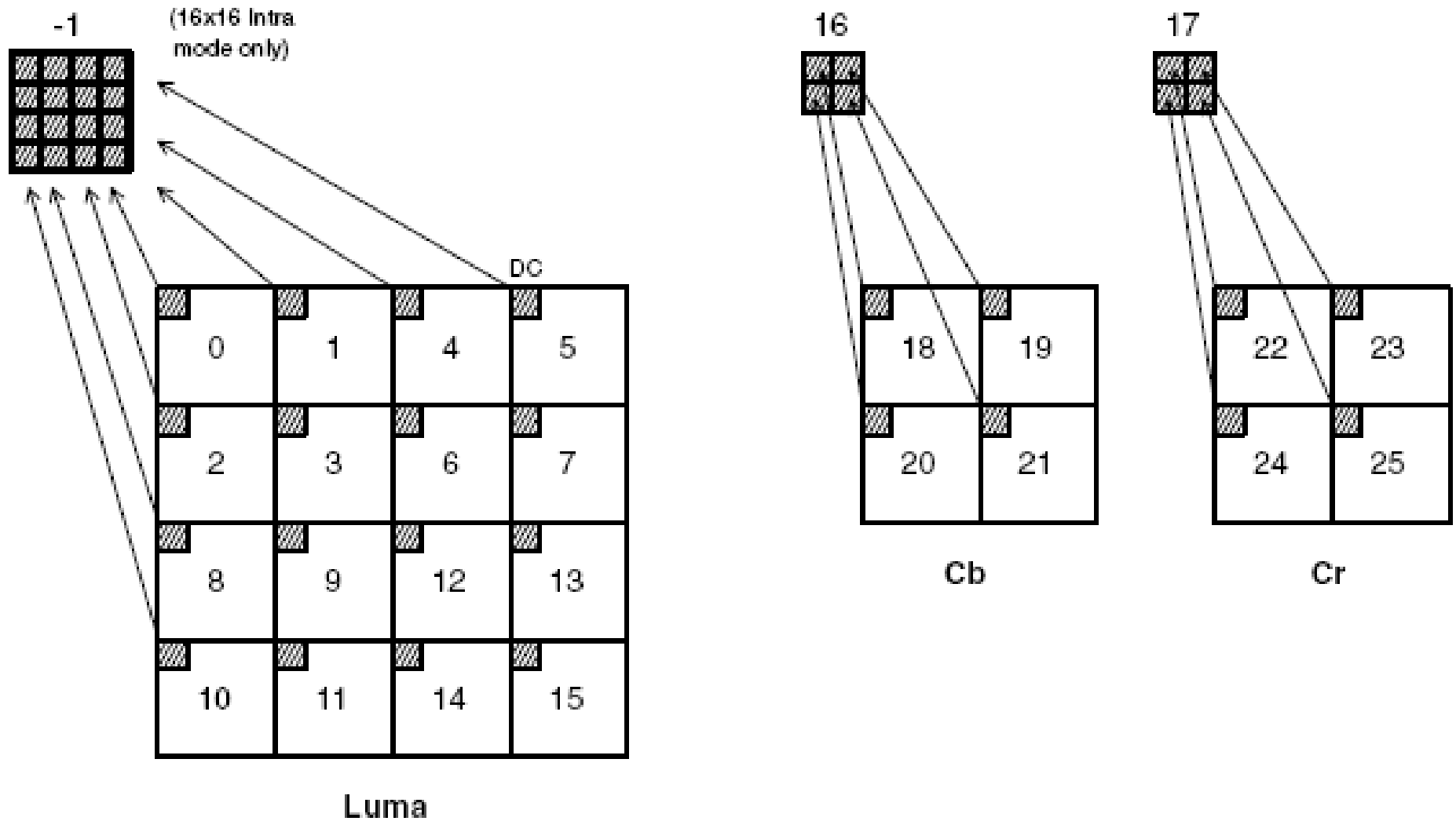
– تبدیل هادامار برای ضرایب DC مولفه‌ی (روشنایی) مود 16×16 ،

– تبدیل هادامار برای آرایه 2×2 از ضرایب DC مؤلفه‌ی (رنگی)

– تبدیل کسینوسی صحیح برای باقی داده‌های بلوک‌های پسماند 4×4 که حاصل از پیش‌بینی inter و یا intra خواهد بود.

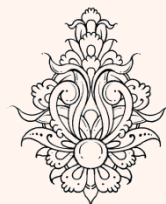


تبدیل کسینوسی صحیح



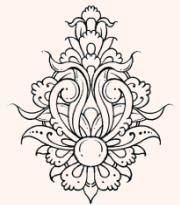
تبدیل کسینوسی (گسسته) صحیح

- تبدیل گسسته کسینوسی صحیح همانند تبدیل DCT معمولی است ولی تفاوت‌هایی بنیادین دارد.
- تبدیل مذکور تبدیلی صحیح است، به گونه‌ای که تمامی عملگرها به صورت صحیح اعمال خواهند شد بدون این که از دقت فرآیند کاسته شود.
- دوم این که به وسیله‌ی استفاده از تبدیل گسسته‌ی کسینوسی در نتیجه‌ی کدگذاری با استفاده از تبدیل و کدگذاری با استفاده از تبدیل معکوس داده‌ای از دست نخواهد رفت.



تبدیل کسینوسی صحیح (ادامه...)

- هسته‌ی مرکزی تبدیل به وسیله‌ی عملیات جمع و جابه‌جایی قابل پیاده‌سازی است.
- به وسیله‌ی ادغام (Scale Multiplication) ضرب مقیاس (به عنوان قسمتی از فرآیند تبدیل) در چندی کننده تعداد عملیات ضرب کاهش می‌یابد.
- پس از اعمال تبدیل برای یک بلوک 4×4 ، ضرایب به صورت زیگ‌زاگ به صورت یک آرایه مرتب می‌شوند.



تبدیل کسینوسی صحیح (ادامه...)

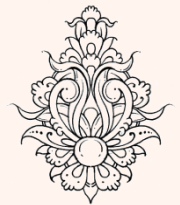
- تبدیل کسینوسی معمولی را می‌توان همانند زیر در نظر گرفت:

$$Y = AXA^T$$

$$Y = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix}$$

که در آن

$$a = \frac{1}{2}, \quad b = \sqrt{\frac{1}{2}} \cos\left(\frac{\pi}{8}\right), \quad c = \sqrt{\frac{1}{2}} \cos\left(\frac{3\pi}{8}\right)$$



$$Y = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix}$$

• رابطه‌ی بالا را می‌توان به صورت زیر بازنویسی کرد:

هسته‌ی تبدیل دو
بعدی

$$Y = (CXC^T) \otimes E$$

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix}$$



$$\begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

$$d = c/b \approx 0.414$$

ضرب نظیر به نظیر



تبدیل کسینوسی صحیح

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \otimes$$

$$\begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes$$

$$\begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix}$$

$$a = \frac{1}{2}, b = \sqrt{\frac{2}{5}}, d = \frac{1}{2}$$



مثال

می‌خواهیم تفاوت تبدیل
کسینوسی صحیح و تبدیل
کسینوسی معمولی را در این مثال
مشاهده کنیم:

\mathbf{X} :

	$j=0$	1	2	3
$i=0$	5	11	8	10
1	9	8	4	12
2	1	10	11	4
3	19	6	15	7

تبدیل معمولی

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T = \begin{bmatrix} 35.0 & -0.079 & -1.5 & 1.115 \\ -3.299 & -4.768 & 0.443 & -9.010 \\ 5.5 & 3.029 & 2.0 & 4.699 \\ -4.045 & -3.010 & -9.384 & -1.232 \end{bmatrix}$$

تفاوت

$$\mathbf{Y} - \mathbf{Y}' = \begin{bmatrix} 0 & 0.079 & 0 & 0.008 \\ -0.295 & -0.868 & -0.664 & 0.190 \\ 0 & 0.341 & 0 & 0.203 \\ 0.224 & 0.190 & -0.055 & 0.868 \end{bmatrix}$$

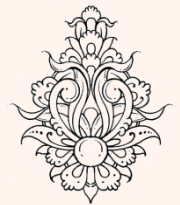
تبدیل کسینوسی
صحیح

$$\mathbf{Y}' = (\mathbf{C}\mathbf{X}\mathbf{C}^T) \otimes \mathbf{E}_f = \begin{bmatrix} 35.0 & -0.158 & -1.5 & 1.107 \\ -3.004 & -3.900 & 1.107 & -9.200 \\ 5.5 & 2.688 & 2.0 & 4.901 \\ -4.269 & -3.200 & -9.329 & -2.100 \end{bmatrix}$$



ادامه...

- بازده فشرده‌سازی در استفاده از تبدیل کسینوسی صحیح تفاوتی با تبدیل معمولی ندارد.
- از مزایای استفاده از تبدیل صحیح پیاده‌سازی هسته‌ی تبدیل تنها با استفاده از حساب اعداد صحیح و شامل اپراتورهای جمع و تفریق و شیفت خواهد بود.
- محاسبات ضرب در $\otimes E_f$ تنها با استفاده از یک فرآیند ضرب برای هر ضریب خواهد بود.



تبدیل هادامار

Hadamard Transform

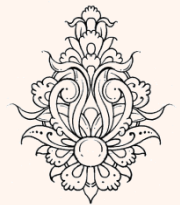
- برای ارسال DC ها از تبدیل هادامار استفاده می‌شود:

$$Y_D = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_{D00} & x_{D01} & x_{D02} & x_{D03} \\ x_{D10} & x_{D11} & x_{D12} & x_{D13} \\ x_{D20} & x_{D21} & x_{D22} & x_{D23} \\ x_{D30} & x_{D31} & x_{D32} & x_{D33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \right) // 2$$

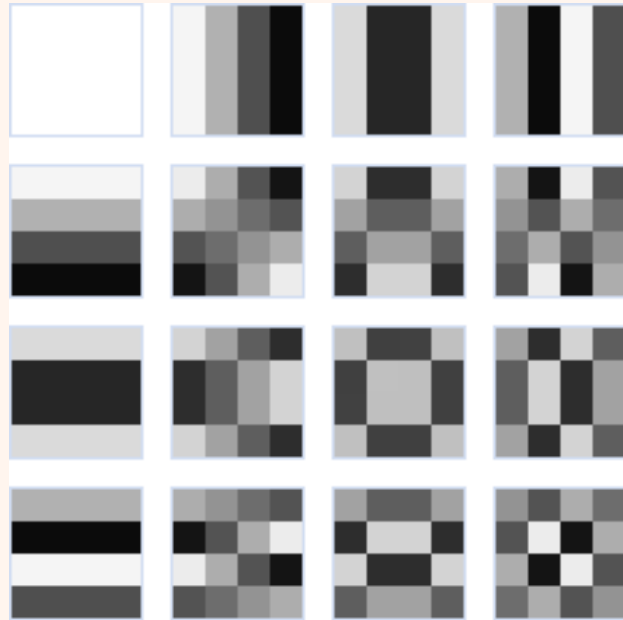
where // = rounding to the nearest integer

- برای مولفه‌های رنگی تبدیل هادامار را برای ماتریس‌های 2×2 خواهیم داشت

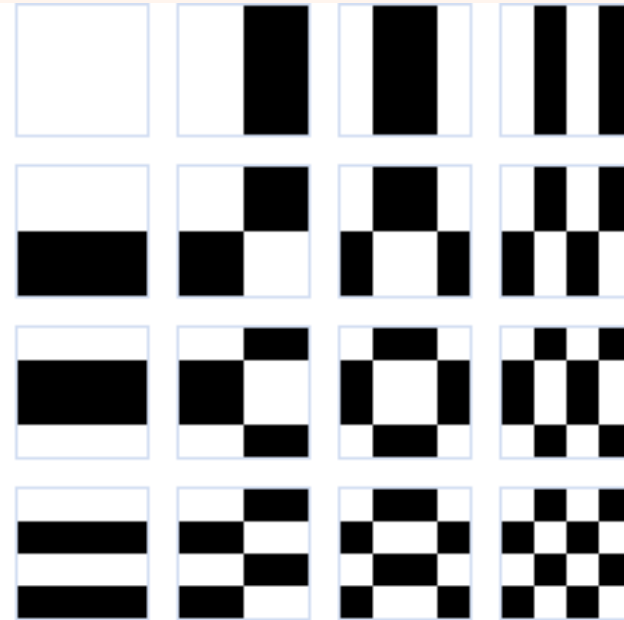
$$\hat{H} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



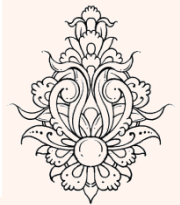
تصاویر پایه تبدیل هادامار



DCT



Walsh-Hadamard



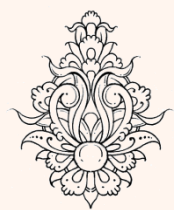
Qstep doubles in size for every increment of 6 in QP

$$Z_{ij} = \text{round}(Y_{ij} / Qstep)$$

در استاندارد ۵۲ مقدار متفاوت برای Qstep است که با QP اندیس گذاری شده است

QP	0	1	2	3	4	5	6	7	8	9	10	11	12
QStep	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	2.5
QP	...	18	...	24	...	30	...	36	...	42	...	48	...
QStep		5		10		20		40		80		160	

• برای صرفه جویی در عملیات فرآیند تخمین مقیاس را با چندی کردن ادغام می کنیم



$$Z_{ij} = \text{round} \left(W_{ij} \cdot \frac{PF}{Qstep} \right)$$

$$W = \bar{C} X C^T$$



$$Z_{ij} = \text{round} \left(W_{ij} \cdot \frac{PF}{Qstep} \right)$$

$$W = \bar{C} X C^T$$

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix}$$

$$\otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix}$$

Position	PF
(0,0), (2,0), (0,2) or (2,2)	a^2
(1,1), (1,3), (3,1) or (3,3)	$b^2/4$
other	$ab/2$

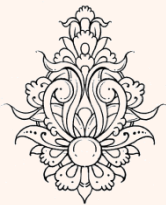
$$Z_{ij} = \text{round} \left(W_{ij} \cdot \frac{MF}{2^{qbits}} \right)$$

• رابطه‌ی مذکور را بازنویسی می‌کنیم:

$$\frac{MF}{2^{qbits}} = \frac{PF}{Qstep}$$

$$qbits = 15 + \text{floor}(QP/6)$$

• که در آن



ادامه...

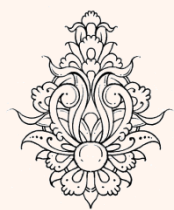
• برای پیاده‌سازی رابطه‌ی مذکور داریم:

$$|Z_{ij}| = (|W_{ij}| \cdot MF + f) \gg qbits$$
$$sign(Z_{ij}) = sign(W_{ij})$$

where \gg indicates a binary shift right. In the reference model software, f is $2^{qbits} / 3$ for Intra blocks or $2^{qbits} / 6$ for Inter blocks.

Table 6.6 Multiplication factor MF

QP	Positions	Positions	Other positions
	(0,0),(2,0),(2,2),(0,2)	(1,1),(1,3),(3,1),(3,3)	
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

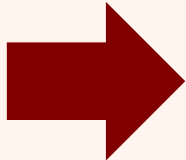


$$\begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \left(\left[\mathbf{x} \right] \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$$

$$Z_{ij} = \text{round}(Y_{ij}/Qstep)$$

Inverse Transform

$$Y'_{ij} = Z_{ij} Qstep$$



$$W'_{ij} = Z_{ij} Qstep \cdot PF \cdot 64$$

the parameter $V = (Qstep.PF.64)$ is defined for $0 \leq QP \leq 5$ and for each coefficient position so that the scaling operation becomes:

Scale

$$W'_{ij} = Z_{ij} V_{ij} \cdot 2^{\text{floor}(QP/6)}$$

Table 6.7 Scaling factor V

QP	Positions (0,0),(2,0),(2,2),(0,2)	Positions (1,1),(1,3),(3,1),(3,3)	Other positions
0	10	16	13
1	11	18	14
2	13	20	16
3	14	23	18
4	16	25	20
5	18	29	23

$$X' = C_i^T W' C_i$$



سپید
بهشتی

$$X'' = \text{round}(X'/64)$$

Final value

فرآیند در یک نگاه

Encoding:

1. Input: 4×4 residual samples: \mathbf{X}
2. Forward 'core' transform: $\mathbf{W} = \mathbf{C}_f \mathbf{X} \mathbf{C}_f^T$
(followed by forward transform for Chroma DC or Intra-16 Luma DC coefficients).
3. Post-scaling and quantisation: $\mathbf{Z} = \mathbf{W} \cdot \text{round}(\text{PF}/\text{Qstep})$
(different for Chroma DC or Intra-16 Luma DC).

Decoding:

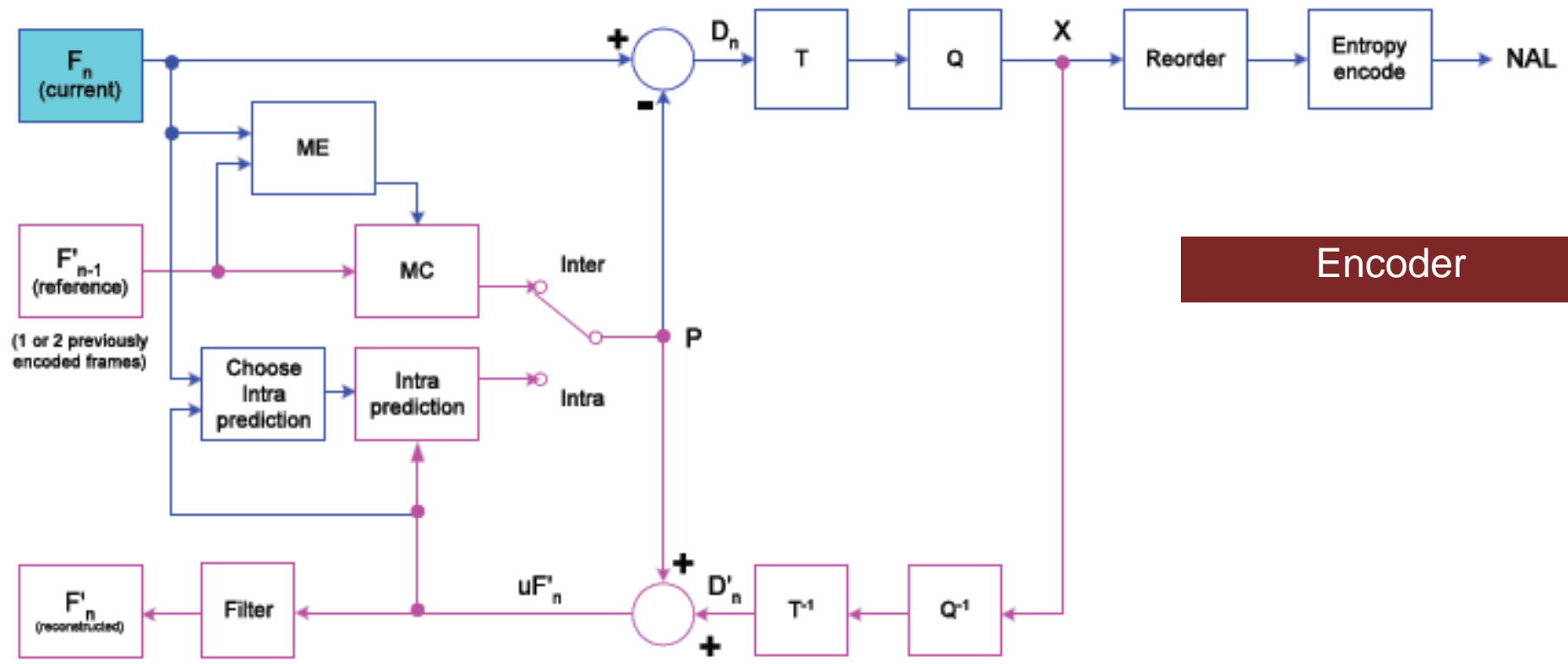
(Inverse transform for Chroma DC or Intra-16 Luma DC coefficients)

4. Decoder scaling (incorporating inverse transform pre-scaling): $\mathbf{W}' = \mathbf{Z} \cdot \text{Qstep} \cdot \text{PF} \cdot 64$
(different for Chroma DC or Intra-16 Luma DC).
5. Inverse 'core' transform: $\mathbf{X}' = \mathbf{C}_i^T \mathbf{W}' \mathbf{C}_i$
6. Post-scaling: $\mathbf{X}'' = \text{round}(\mathbf{X}'/64)$
7. Output: 4×4 residual samples: \mathbf{X}''

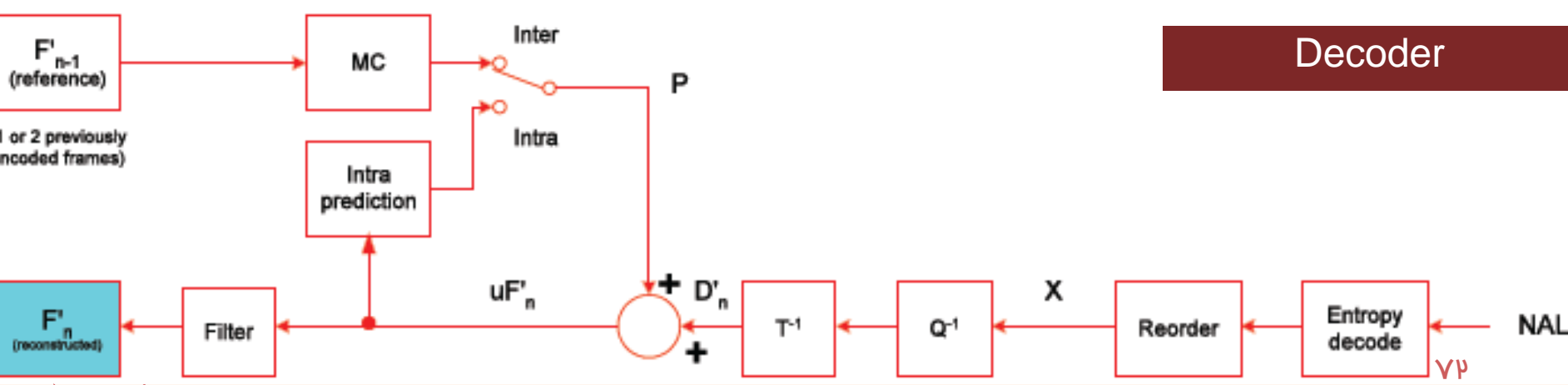
Inverse Transform

$$\begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \left(\left[\mathbf{X} \right] \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$$



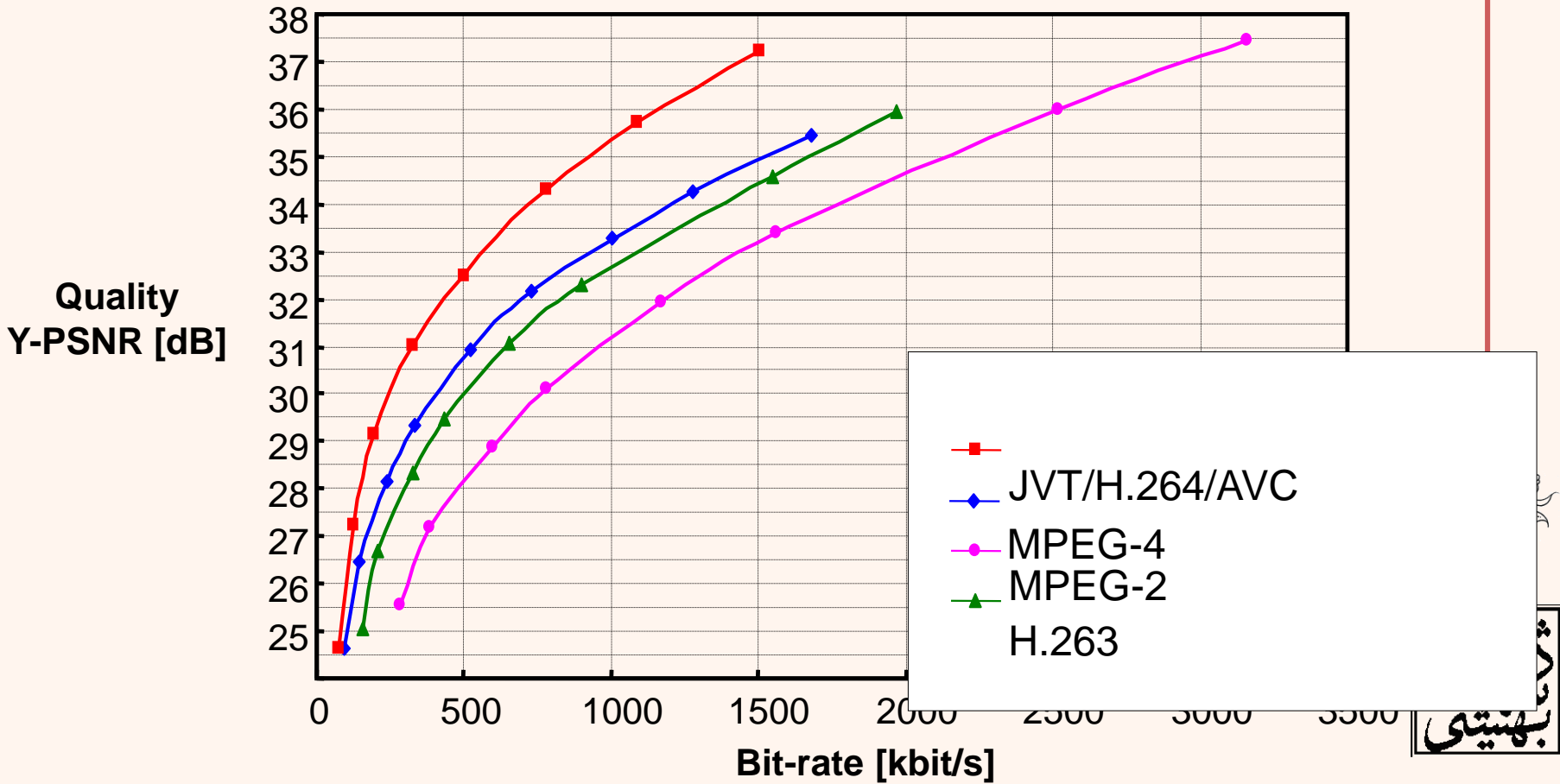


Encoder



Decoder

Tempete CIF 30Hz



بهبودی

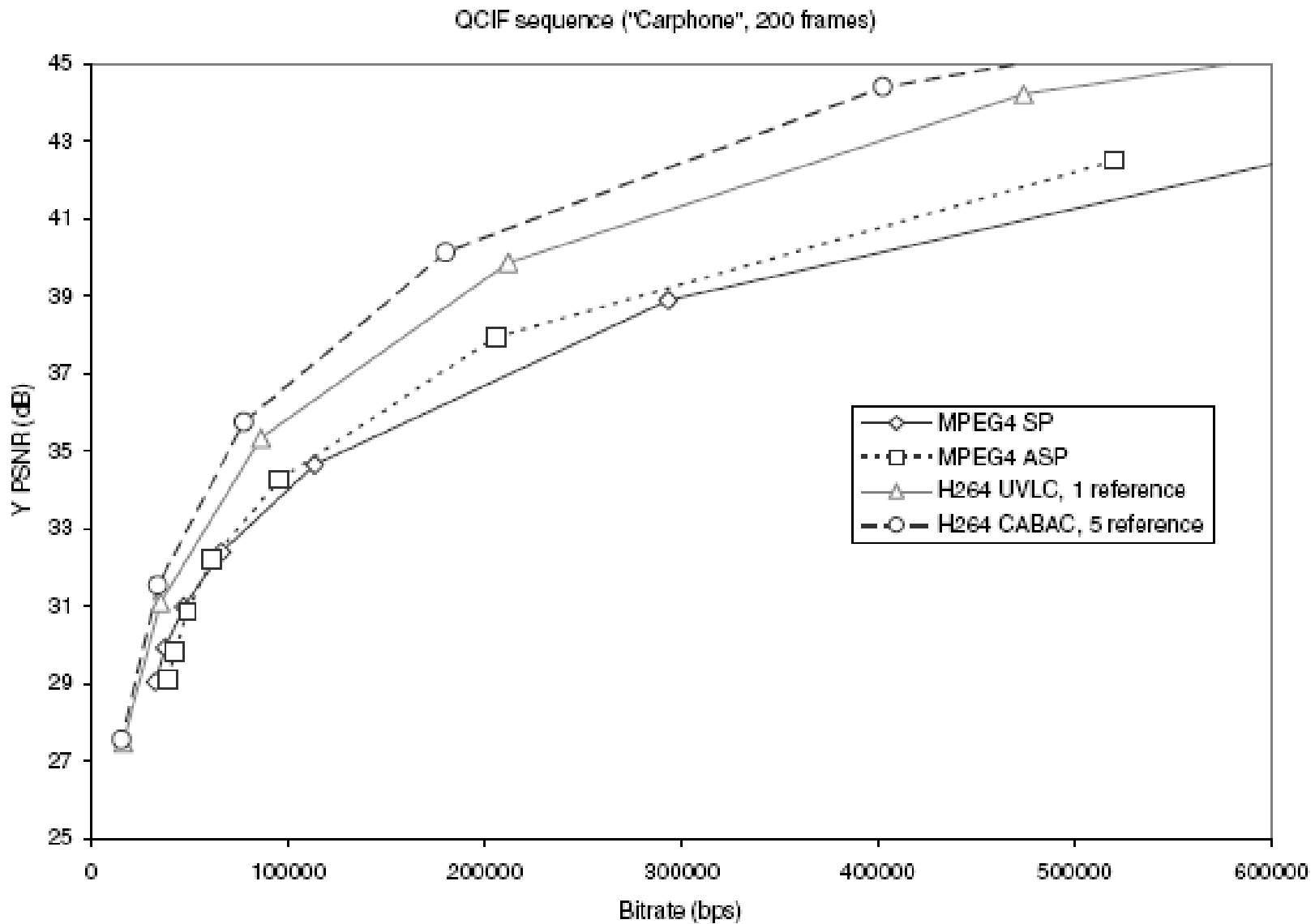


Figure 7.34 Rate-distortion comparison: 'Carphone', QCIF



H.264/AVC Software Coordination

Current software version: JM 18.6

JM Software

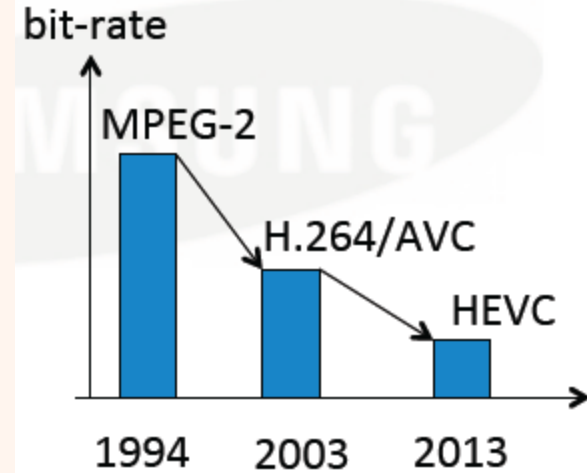
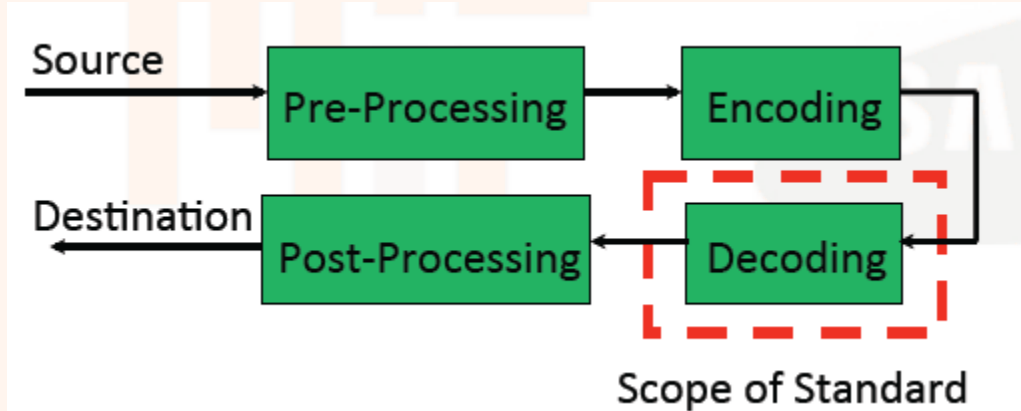
- ▶ Download the latest H.264/AVC reference software
- ▶ Release notes (Bug Tracker)
- ▶ Fixed bugs (Bug Tracker)

KTA Software

VCEG historically developed some software known as the KTA software during the period starting in early 2005 that led up to the launch of the [HEVC standardization project](#) in 2010. There were two versions of the KTA software that were developed during that time: One based on JM 11.0 and one based on JM 14.2. These version contain **different** coding tools. See the included changelog for details.

- Start
- News
- Organisation
- Competences
- Fields of Application
- Alliances & Committees
- Products
- Events
- Staff**
- Jobs
- Visitors
- Contact
- HHI Home

H.265/HEVC

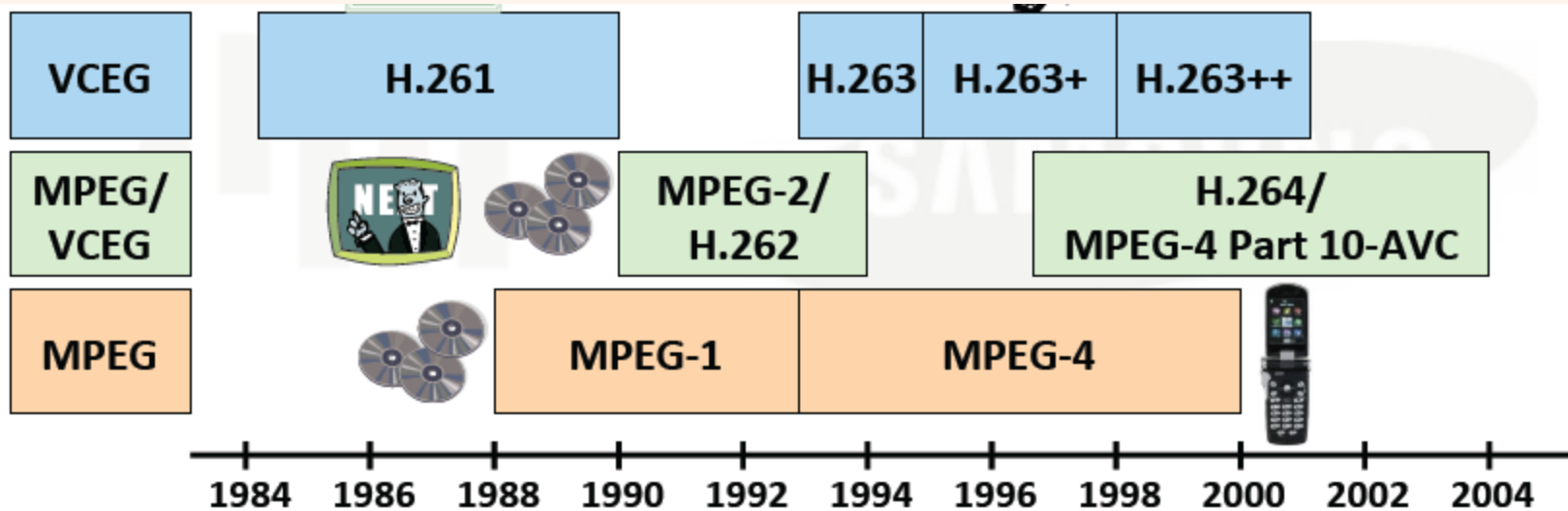


~2x improvement in compression every decade

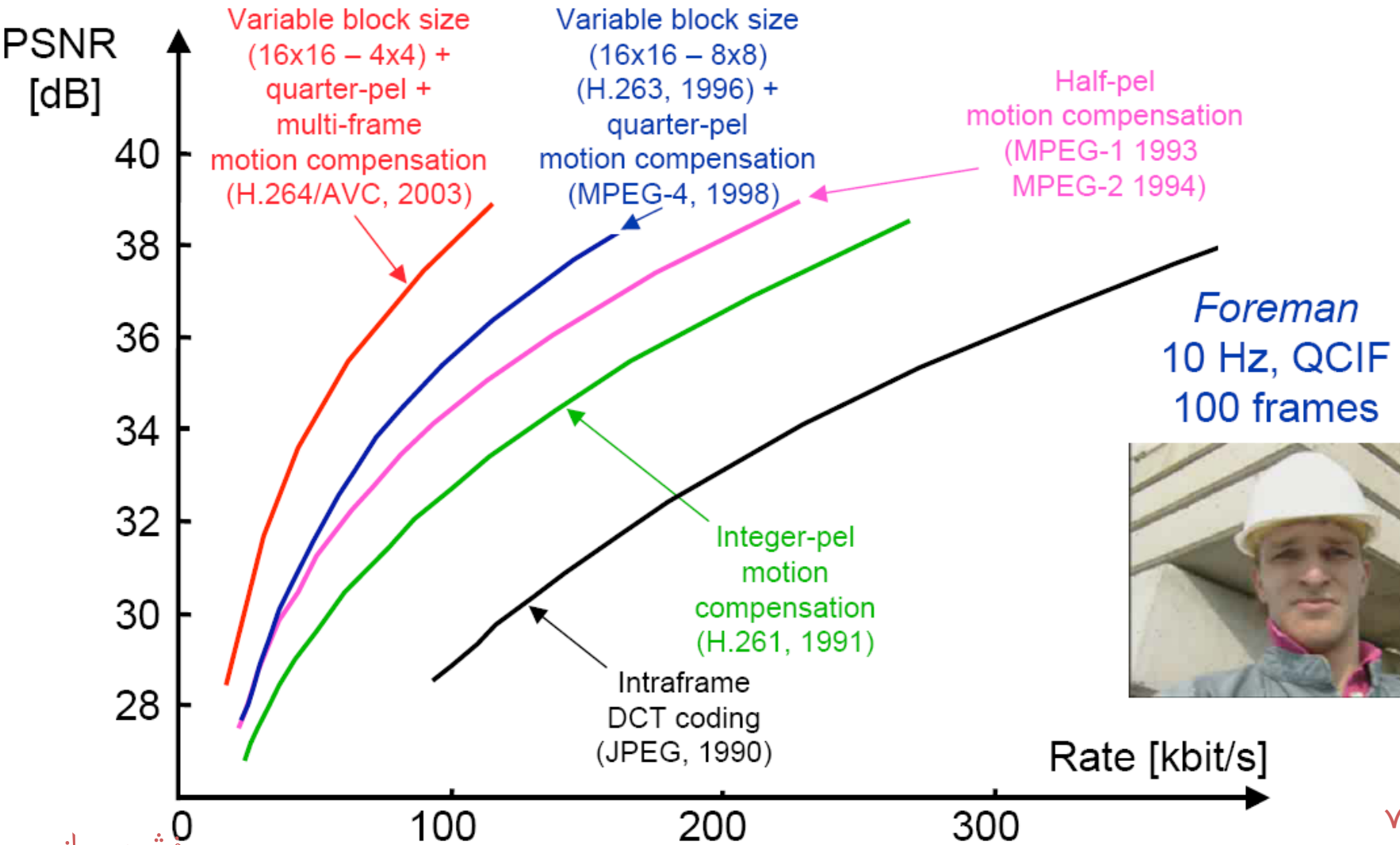


H.265/HEVC

- MPEG: Moving Picture Experts Group (ISO/IEC)
- VCEG: Video Coding Experts Group (ITU-T)
- Other standards: VC1, VP8/VP9, China AVS, RealVideo

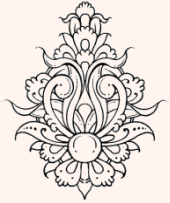
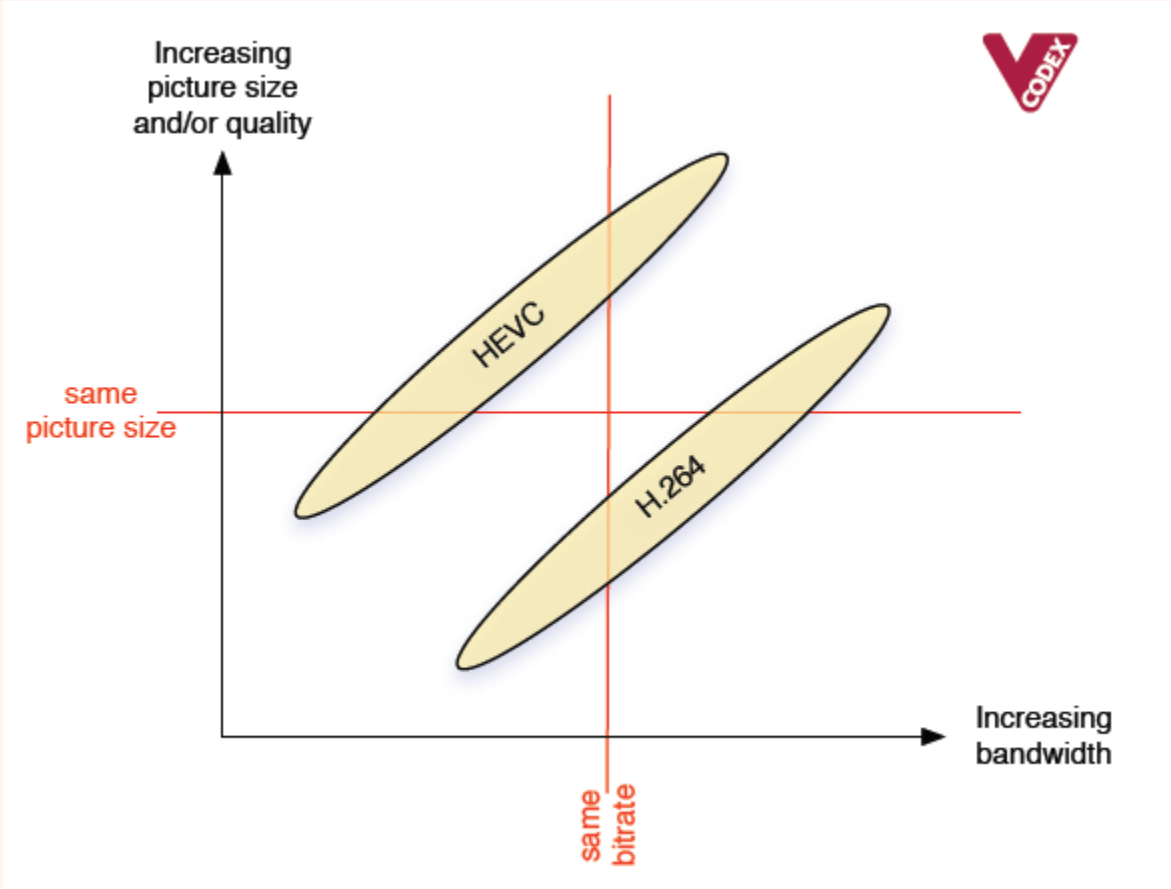


مقایسه



High Efficiency Video Coding

How It HEVC (H.265) Works



Achieves 2x higher compression compared to H.264/AVC

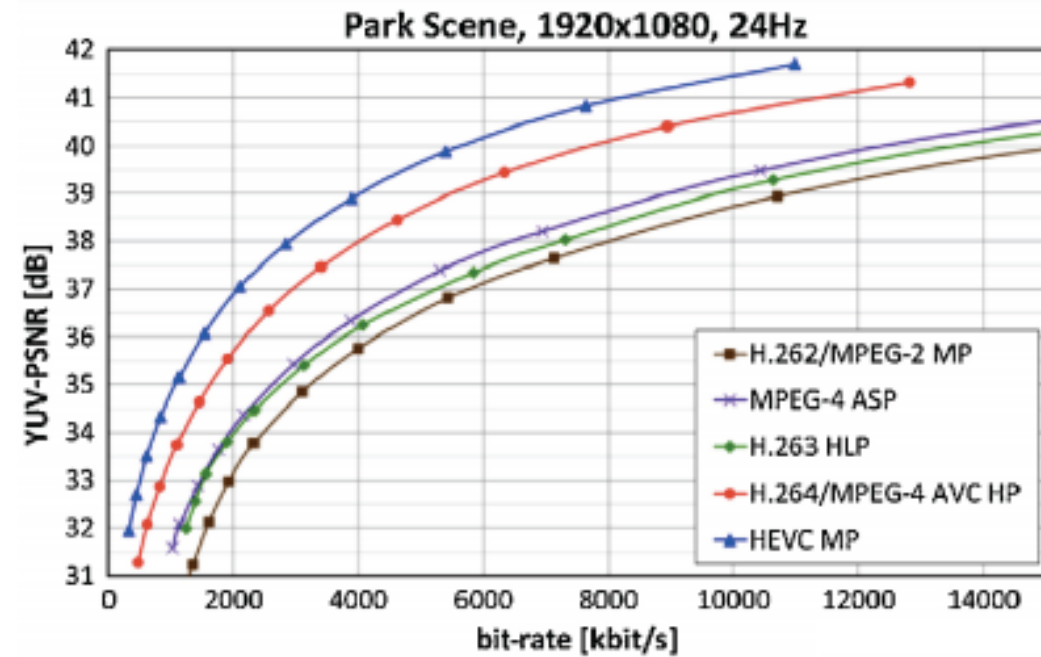
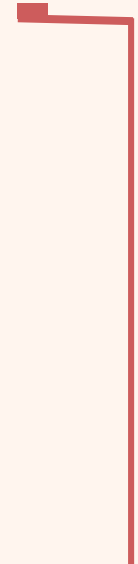
High throughput (Ultra--HD 8K @ 120fps) & low power –
Implementation friendly features (e.g. built--inparallelism)

Easier streaming of HD video to mobile devices

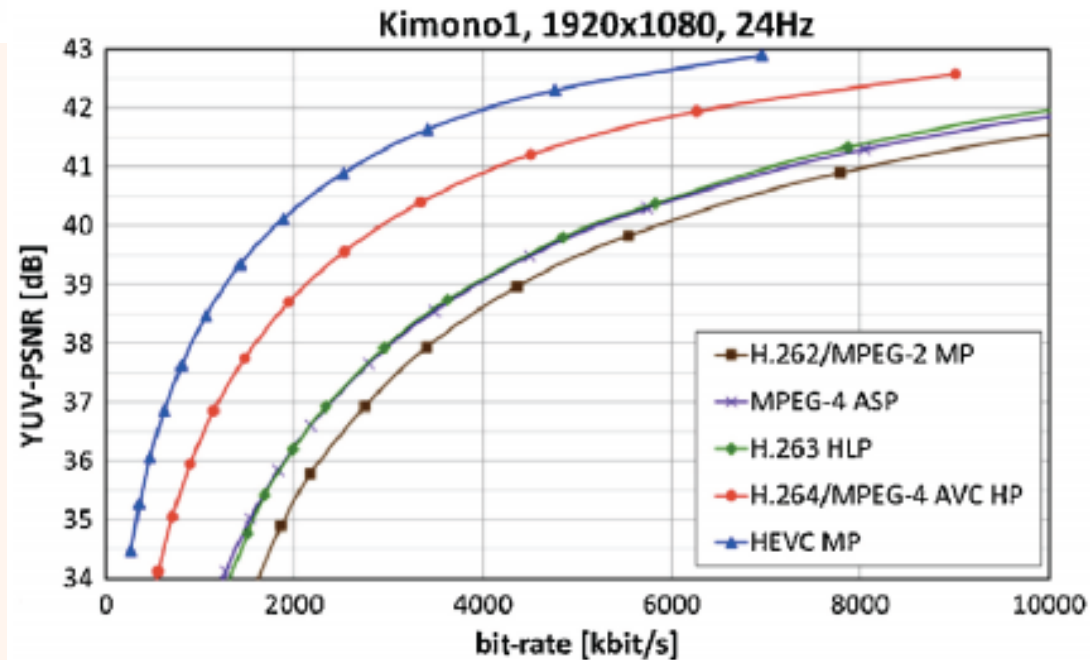
*“HEVC will provide a flexible, reliable and robust solution, future--
-proofed to support the next decade of video” --- ITU---T Press Release
(2013)*



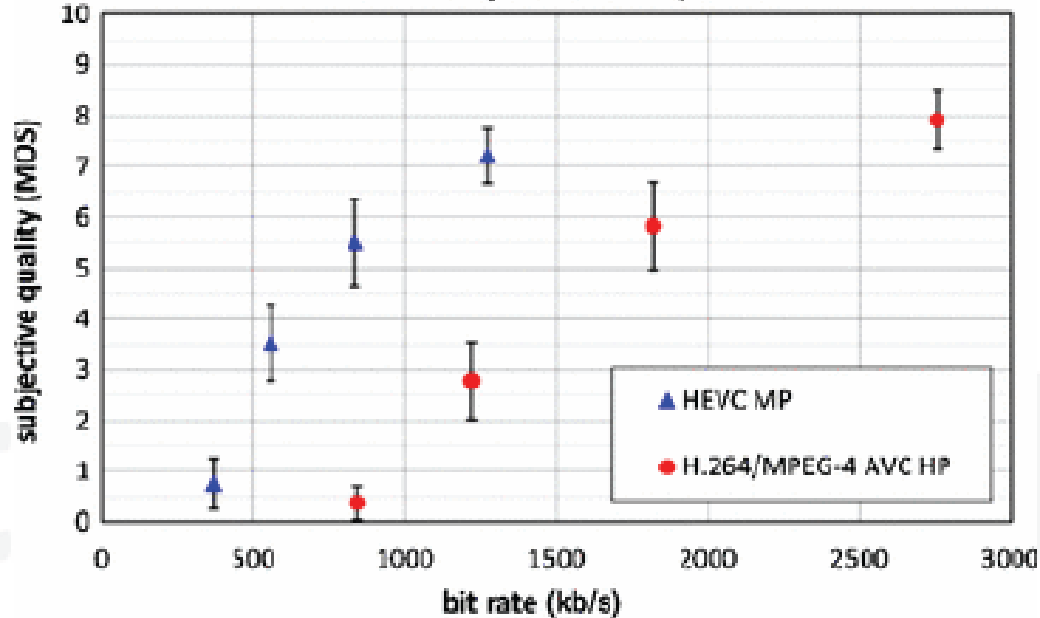
مقایسه



J. R. Ohm et al., "Comparison of the Coding Efficiency of Video Coding Standards—Including High efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, 2012



Kimono1, 1920x1080, 24Hz

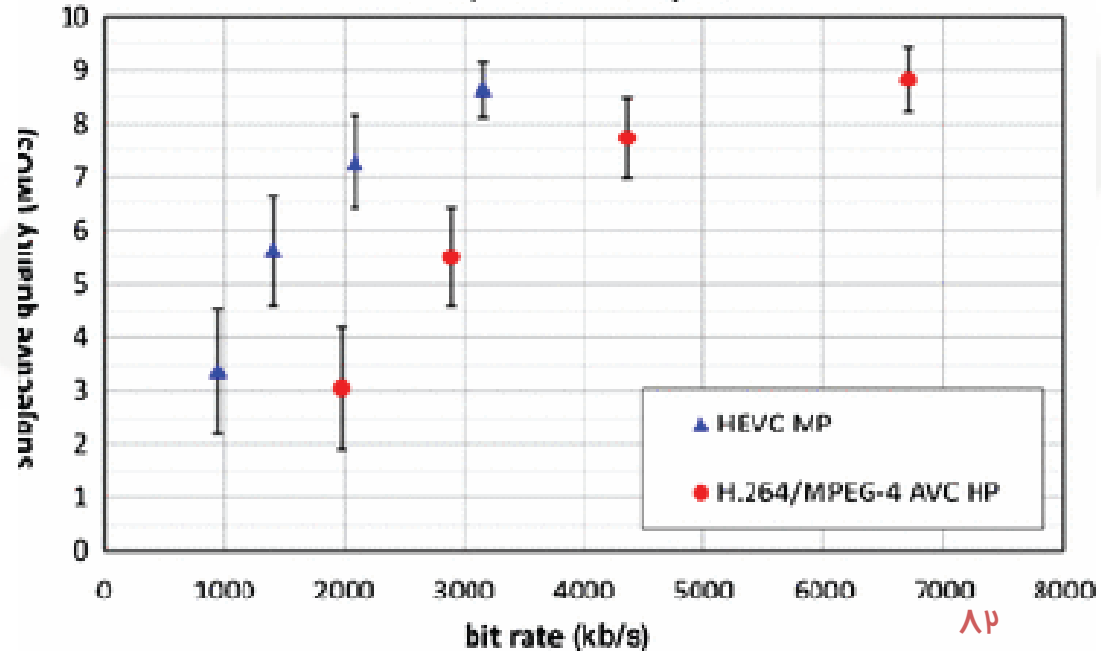


Subjective Evaluation



J. R. Ohm et al., "Comparison of the Coding Efficiency of Video Coding Standards—Including High efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, 2012

Cactus, 1920x1080, 50Hz

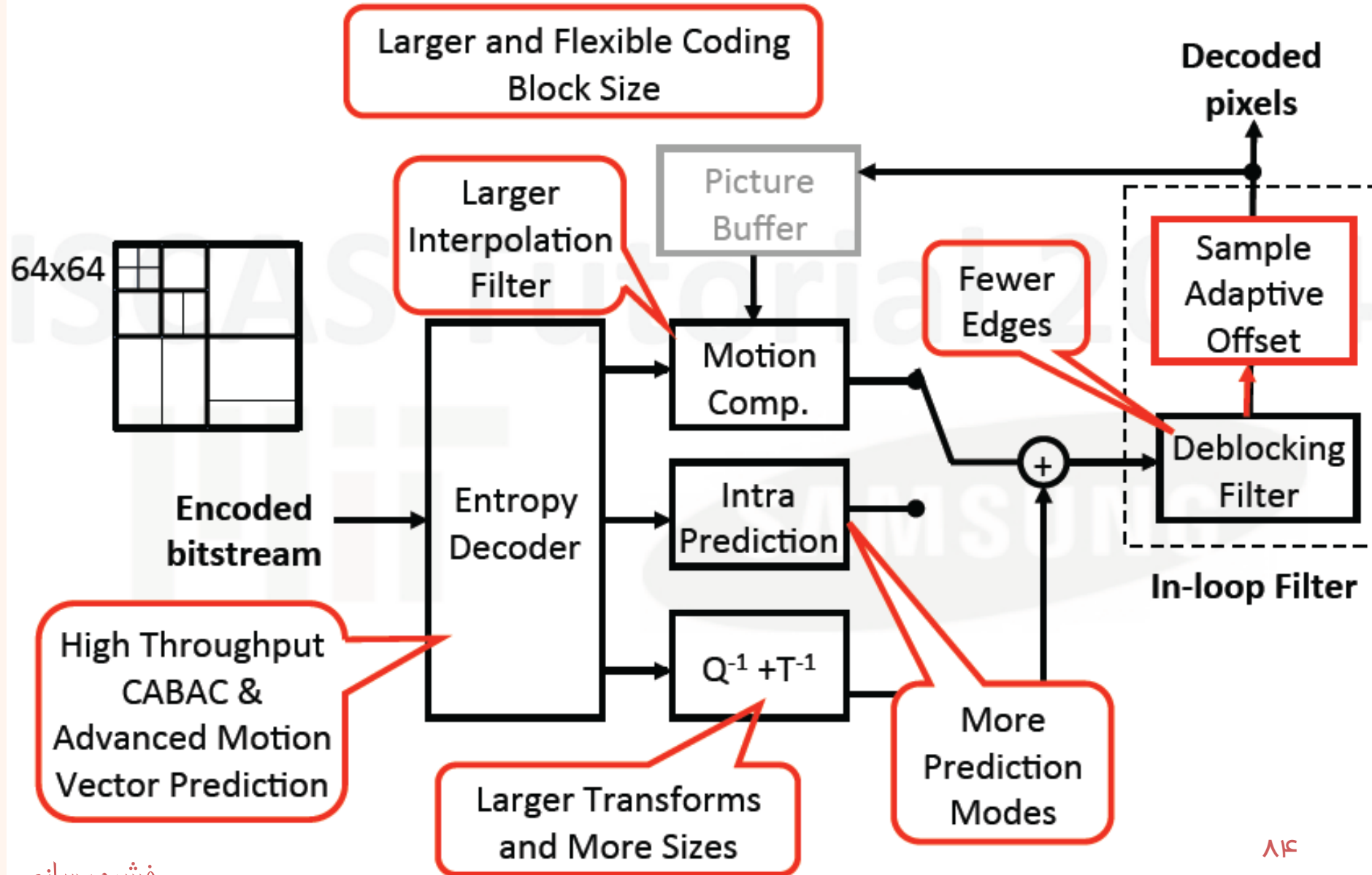


مهم‌ترین تغییرات

https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

- همانند H.264 استاندارد H.265 نیز از بلوک با اندازه‌ی انعطاف پذیر استفاده می‌نماید.
- همچنین از روش‌های پیش‌بینی متعددی برای به دست آوردن بازده بالاتر سود می‌جوید.
- موارد جدید در این استاندارد:
 - اندازه بلوک بزرگتر
 - ابزار برای کدگذاری و کدگشایی موازی
 - بهبود کدگذاری اطلاعات حرکتی
 - اسلوب‌های intra بیشتر
 - تبدیل‌های انعطاف پذیر بر اساس اندازه بلوک
 - بهبود پردازش in-loop
 - استفاده از quad-tree بهبود یافته

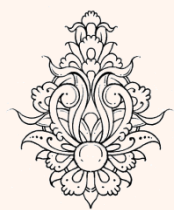




HEVC Demystified: A Primer on the H.265 Video Codec

COMPONENT	MPEG-2	H.264	HEVC/H.265
General	Motion compensated predictive, residual, transformed, entropy coded	Same basics as MPEG-2	Same basics as MPEG-2
Intra prediction	DC Only	Multi-direction, multi-pattern, 9 intra modes for 4x4, 9 for 8x8, 4 for 16x16	35 modes for intra prediction, 32x32, 16x16, 8x8 and 4x4 prediction size
Coded Image Types	I, B, P	I, B, P, SI, SP	I, P, B
Transform	8x8 DCT	8x8 and 4x4 DCT-like Integer Transform	32x32, 16x16, 8x8 and 4x4 DCT-like Integer Transform
Motion Estimation Blocks	16x16	16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4	64x64 and hierarchical quad-tree partitioning down to 32x32, 16x16, 8x8 Each size can be partitioned once more in up to 8 ways
Entropy Coding	Multiple VLC tables	Context adaptive binary arithmetic coding (CABAC) and context adaptive VLC tables (CAVLC)	Context adaptive binary arithmetic coding (CABAC)
Frame Distance for Prediction	1 past and 1 future reference frame	Up to 16 past and/or future reference frames, including long-term references	Up to 15 past and/or future reference frames, including long-term references
Fractional Motion Estimation	½ pixel bilinear interpolation	½ pixel 6-tap filter, ¼ pixel linear interpolation	¼ pixel 8-tap filter
In-Loop Filter	None	Adaptive deblocking filter	Adaptive deblocking filter and sample adaptive offset filter

(High profiles only)



- G. J. Sullivan, et al. "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, 2012
- V. Sze, M. Budagavi, G. J. Sullivan (Editors), "High Efficiency Video Coding (HEVC): Algorithms and Architectures," Springer, 2014

