Image and Video compression basic

图像视频编码基本原理 和技术相关介绍

But life is short and information endless‥ ‥ ‥

Abbreviation is a necessary evil and the abbreviator's business is to make the best of a job which, although intrinsically bad, is still better than nothing.

Aldous Huxley

1、compression basic

2、 Inter predictions Basic

3、 H. 264 standard

4、Newest standard HEVC

彩色图像的压缩

图b是图a压缩了230倍后又恢复的图像.

a b
c d

FIGURE 6.51
Color image compression. (a) Original RGB image. (b) Result of compressing and decompressing the image in (a).

Image, Video and compression basic

- Image — matrix:

- Video:

- Micro-block (MB)— small piece of image:

Introduction of still image compression：

Compression is one of the key parts for multimedia technologies，which purpose is try to use as small as possible data to represent the same information.

Images with different formats have to use different compression algorithms. We use, in general, a matrix to represent an image and the characteristics of elements in the matrix are related to the image types.

The purpose of compression is to remove redundancies from image data, including coding redundancy, inter-pixel redundancy and psycho-visual redundancy.

What do we know about compression or coding?

- Huffman coding
- Run-length coding
- (adaptive) arithmetic coding
- Quantization? Transformation?

What kind of data will be good for above codec?

What do we do if the data is not good for the codec?

Introduction of still image compression：

（1）Basic image types

- **Black and white：Image pixels are represented by 0 and 1（or 0 and 255）and preferred the former for the sake of compression.**

- **Gray level images：Generally there are 2^8 gray levels and represented by 256 integers from 0 to 255.（Medical or satellite images may have more gray levels than the normal, such as 2^{12} or 2^{16})**

- **Color images：represented by (R, G, B) planes in general, each plane has 256 intensity levels. Question：index images，such as GIF format？**

（2）Format of color images

- **RGB format：** Tricolor （the basic representation）
- **YUV format：** Y—Luminance (Intensity), UV—Chroma component . YUV are often used in image/video compression. The relationship between RGB and YUV is:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

2.03U=B－ Y

1.14V=R － Y

色差信号

What do you see from each row above?

- There are several other formats, such as HSI, YIQ, YCbCr etc., used in different applications. However, all of them are transformed from RGB format. Try to build a transformation yourself？ How about integer transform?
- **Index format：** such as GIF, part of PNG and so on.

（3）图像压缩的基本组成部分：

压缩的主要目的是去除冗余，包括统计冗余（编码冗余）、空间冗余和心理视觉冗余。

主要组成部分：变换—量化—编码

- 变换：（理论上）1—1 的数学变换，把数据变成更有利于压缩的形式，例如能量更为集中、信息熵更低等等。常见的变换有彩色模型的变换，如RGB→YUV、YCbCr；平面间的变换，如FFT、DCT、DWT等。
- 量化：多对一的变换，有损压缩的主要来源。相当于去除心理视觉冗余（数码相机中调整图像质量的参数与此有关）。
- 编码：把字符流变成二进制位流。（包括对字符的重新排序等）

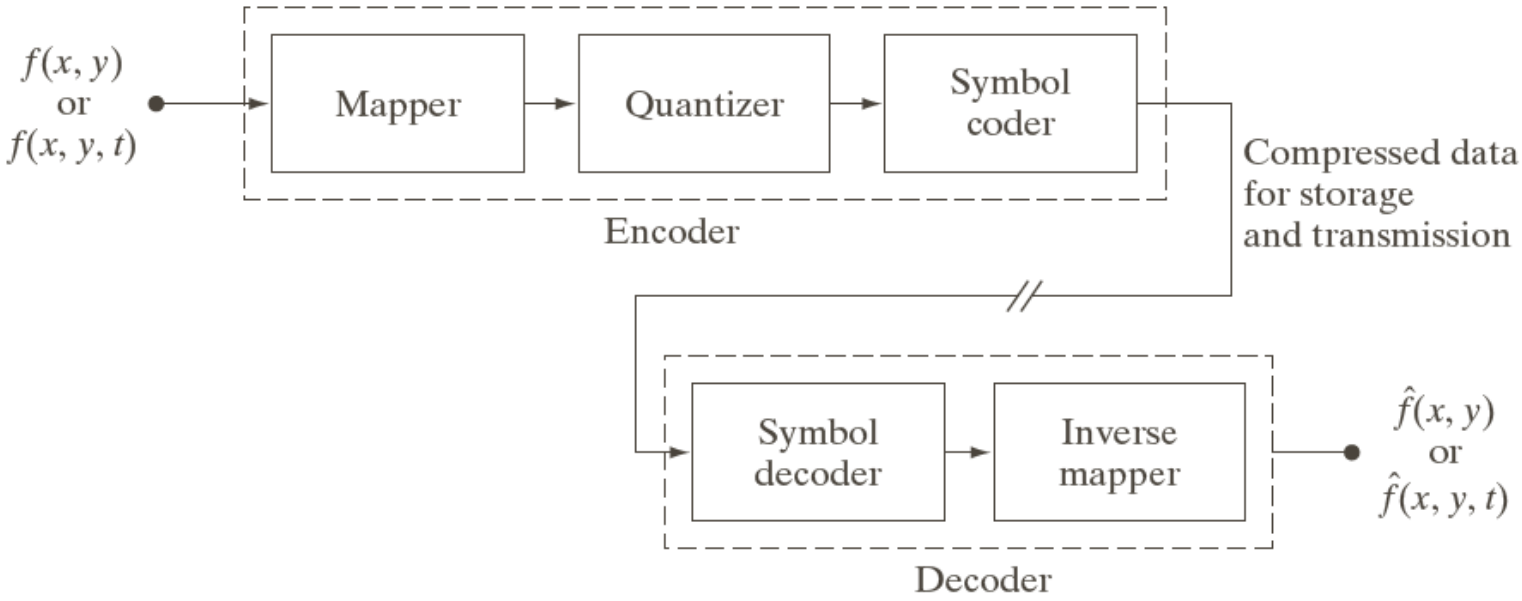
以上选择的各种组合可以构成不同的压缩系统。压缩的基本类型：

- 无损：去除统计冗余，方法中一般只包括变换（？）和编码，不用量化器
- 有损：同时去除统计和心理视觉冗余，使用量化器

a b c

FIGURE 8.1 Computer generated 256 × 256 × 8 bit images with (a) coding redundancy, (b) spatial redundancy, and (c) irrelevant information. (Each was designed to demonstrate one principal redundancy but may exhibit others as well.)

FIGURE 8.5
Functional block diagram of a general image compression system.



Begin with Huffman coding & Lossless compression

- 熵编码之一：**Huffman Coding (Variable-Length Coding,** 变长编码**)**
- **What kind of data will be good for Huffman coding?**
- **What do we do if the data is not good for the codec?**

Variable-Length Coding （变长编码）

目的是去除统计冗余

先看看霍夫曼（Huffman）编码：首先，Huffman编码实际上是一个确定符号码字的过程，当字母集合中只有两个字符时，我们可以用0和1来分别表示。当字符的个数超过2时，则逐步将概率小的的字符合并成组，最后归结为两个大组，然后再分解成一个树型结构，每个字符在树上的位置就确定了它对应的码字。

通过一个例子说明Huffman编码的过程：A：0.5，b：0.25，c：0.125，d：0.125

A 0.5 B 0.25 C 0.125 D 0.125	A 0.5 B 0.25 {c,d} 0.25	A 0.5 {b,{c,d}} 0.5	{a,{b,{c,d}}} 1.0

Variable-Length Coding （变长编码）（Cont.）

A 0.5 B 0.25 C 0.125 D 0.125	A 0.5 B 0.25 {c,d} 0.25	A 0.5 {b,{c,d}} 0.5	{a,{b,{c,d}}} 1.0

由上述分解形成一棵二值树，并由此给定了每个字符的码字：

a：0， b：10， c：110， d：111

Huffman编码不需要特别的终止符，给定一二进制串，沿树查找，就可以找到对应的所有字符。当符号概率是2的幂次方时，Huffman编码产生的结果最好的（和信源的熵是一致）。

选作作业：a:0.2， e:0.3， o:0.2， i:0.2,u:0.1。字符串：eiaiouoou。尝试用Huffman编码解码。

再看一个例子：假设一幅图像有6个灰度级别，概率分布如表所示

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

FIGURE 8.11 Huffman source reductions.

信源化简

Huffman编码分配过程（从上向下按概率大小分配0和1

Original source			Source reduction			
Sym.	Prob.	Code	1	2	3	4
a_2	0.4	1	0.4 1	0.4 1	0.4 1	0.6 0
a_6	0.3	00	0.3 00	0.3 00	0.3 00	0.4 1
a_1	0.1	011	0.1 011	0.2 010	0.3 01	
a_4	0.1	0100	0.1 0100	0.1 011		
a_3	0.06	01010	0.1 0101			
a_5	0.04	01011				

这个例子中编码的平均长度是

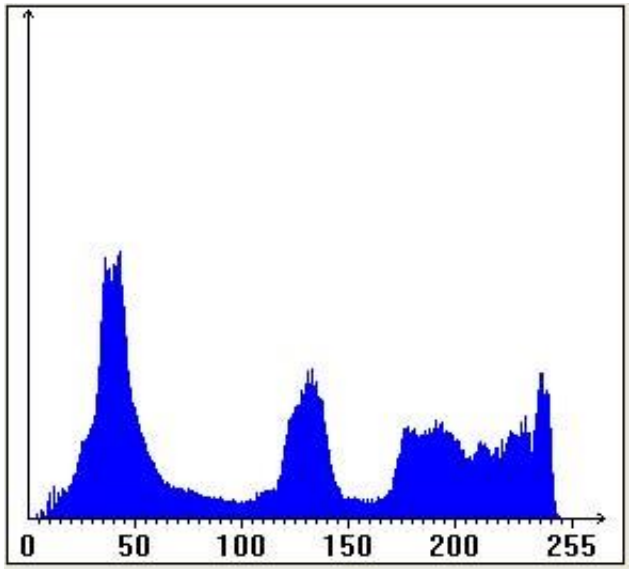
$$L_{avg}=(0.4)(1)+(0.3)(2)+(0.1)(3)+(0.1)(4)+(0.06)(5)+(0.04)(5)=2.2 \text{ bit/symbol}$$

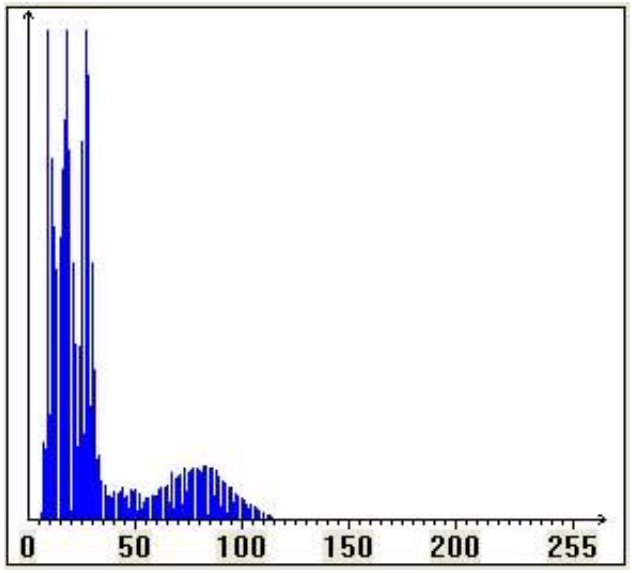
Error-Free compression

Source symbol	Probability	Binary Code	Huffman	Truncated Huffman	B_2-Code	Binary Shift	Huffman Shift
Block 1							
a_1	0.2	00000	10	11	C00	000	10
a_2	0.1	00001	110	011	C01	001	11
a_3	0.1	00010	111	0000	C10	010	110
a_4	0.06	00011	0101	0101	C11	011	100
a_5	0.05	00100	00000	00010	C00C00	100	101
a_6	0.05	00101	00001	00011	C00C01	101	1110
a_7	0.05	00110	00010	00100	C00C10	110	1111
Block 2							
a_8	0.04	00111	00011	00101	C00C11	111 000	00 10
a_9	0.04	01000	00110	00110	C01C00	111 001	00 11
a_{10}	0.04	01001	00111	00111	C01C01	111 010	00 110
a_{11}	0.04	01010	00100	01000	C01C10	111 011	00 100
a_{12}	0.03	01011	01001	01001	C01C11	111 100	00 101
a_{13}	0.03	01100	01110	10 0000	C10C00	111 101	00 1110
a_{14}	0.03	01101	01111	10 0001	C10C01	111 110	00 1111
Block 3							
a_{15}	0.03	01110	01100	10 0010	C10C10	111 111 000	00 00 10
a_{16}	0.02	01111	010000	10 0011	C10C11	111 111 001	00 00 11
a_{17}	0.02	10000	010001	10 0100	C11C00	111 111 010	00 00 110
a_{18}	0.02	10001	001010	10 0101	C11C01	111 111 011	00 00 100
a_{19}	0.02	10010	001011	10 0110	C11C10	111 111 100	00 00 101
a_{20}	0.02	10011	011010	10 0111	C11C11	111 111 101	00 00 1110
a_{21}	0.01	10100	011011	10 1000	C00C00C00	111 111 110	00 00 1111
Entropy	4.0						
Average length		5.0	4.05	4.24	4.65	4.59	4.13

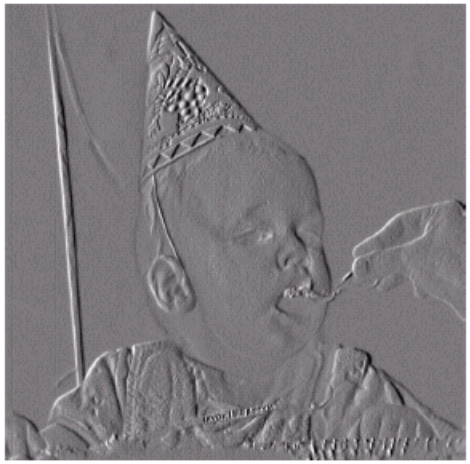
TABLE 8.5 Variable-length codes.

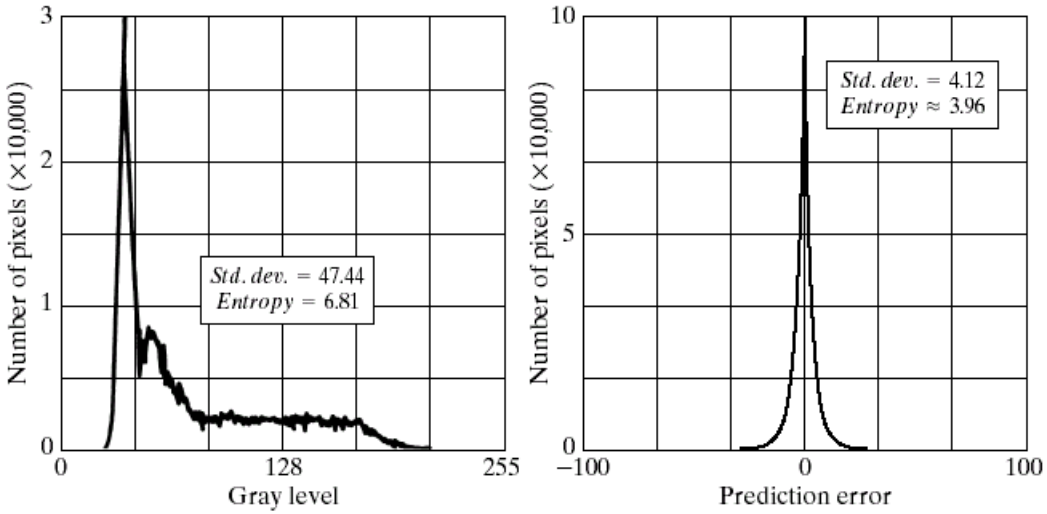
- **What kind of data will be good for Huffman coding?**





- **What do we do if the data is not good for the codec?**





变换能够调整图像的直方图

无损压缩

无损压缩主要由可逆变换和熵编码两部分组成。现介绍一些信息论的基本概念：

设符号集A中含有有限个（M个）符号，如$A=\{a_1, a_2, \cdots\cdots, a_M\}$（如灰度级别）。信源(图像)X可以看成是一个离散的随机过程（即一个随机变量序列X_i，i=1，……），形式为X_1，X_2，……，其中每个随机变量X_i在字母A集中取值。

概率：在一个有限集合内（有限维图像），$a_i \in A$出现的概率$p(a_i)=p_i$简单说就是符号a_i出现的次数除以元素的总个数。相当于normalized直方图。

熵：图像信息量的量度。与符号表示的信息内容和符号的不可预测或不可想象程度有关。如果符号出现的概率低，那么它所传送的信息量就比出现概率高的符号要大。这种定量的概念可以用下列关系表示

$$I(a_i) = \log_2(1/\ p(a_i)) \qquad a_i \in A$$

易见，概率越大熵越小，符号所携带的信息量也就越少。

带有符号集A的信源X（例如X可以是一幅图像）的熵H(X)定义成信源中每个符号的平均信息量，表示为

$$H(X)=\sum_{a\in A}p(a)\log_2(1/p(a))=-\sum_{a\in A}p(a)\log_2 p(a)$$

符号概率分布的越不均匀，信源的熵越小。符号的概率均匀分布，即当所有符号等概率分布时，则熵最大。

定理：对一组图像数据编码后获得的平均比特不会小于这幅图像的熵。

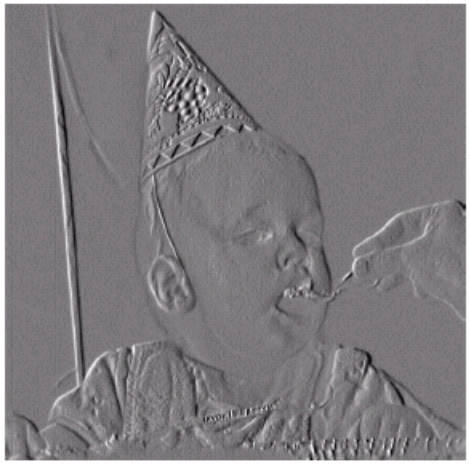
思考题：任意找一副自然图像，（1）计算这一图像的熵和直方图；（2）用后一像素减前一像素得到另一幅图像，计算这幅图像的熵和直方图。（3）对图像作Haar小波变换，计算变换后图像的熵。

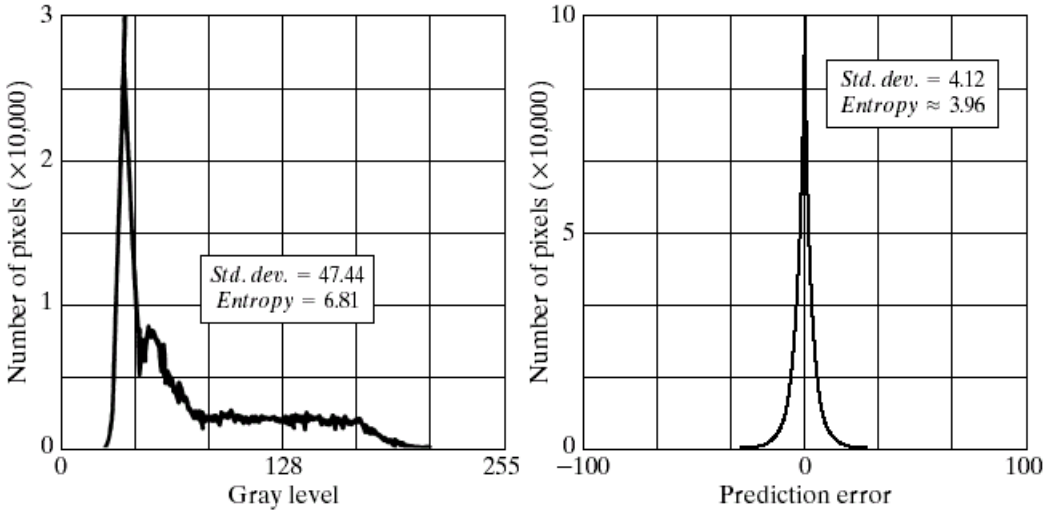
你想到了什么？

Error-Free compression

a
b c

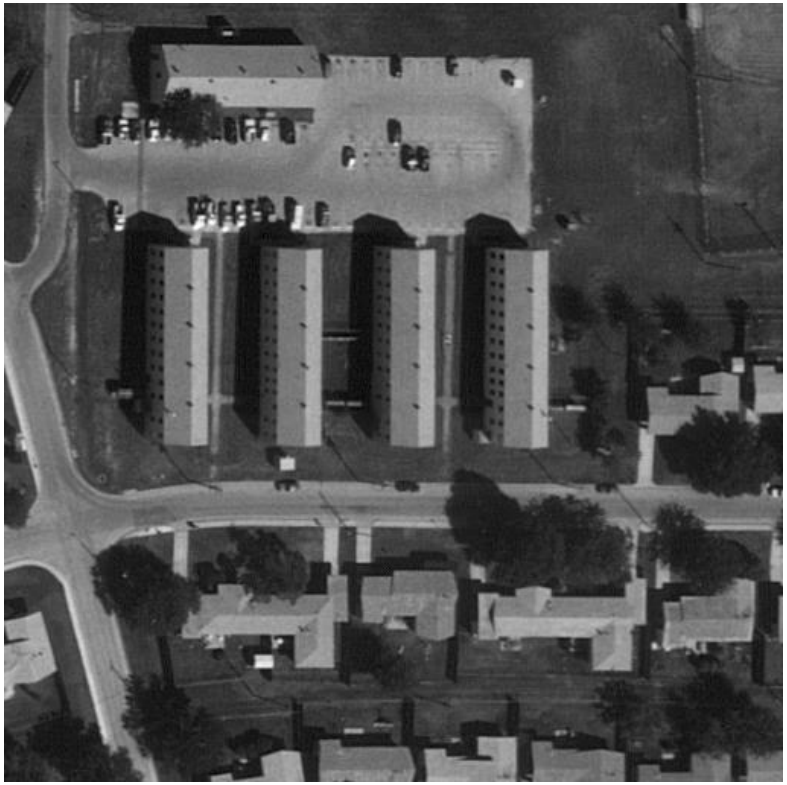
FIGURE 8.20
(a) The prediction error image resulting from Eq. (8.4-9).
(b) Gray-level histogram of the original image.
(c) Histogram of the prediction error.

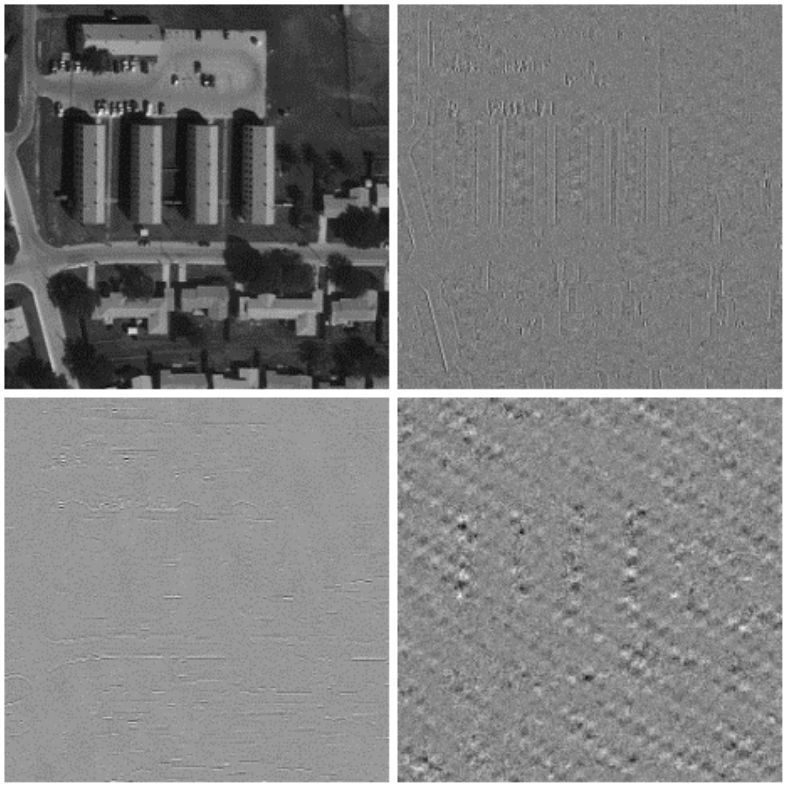




变换能够调整图像的信息熵

Error-Free compression



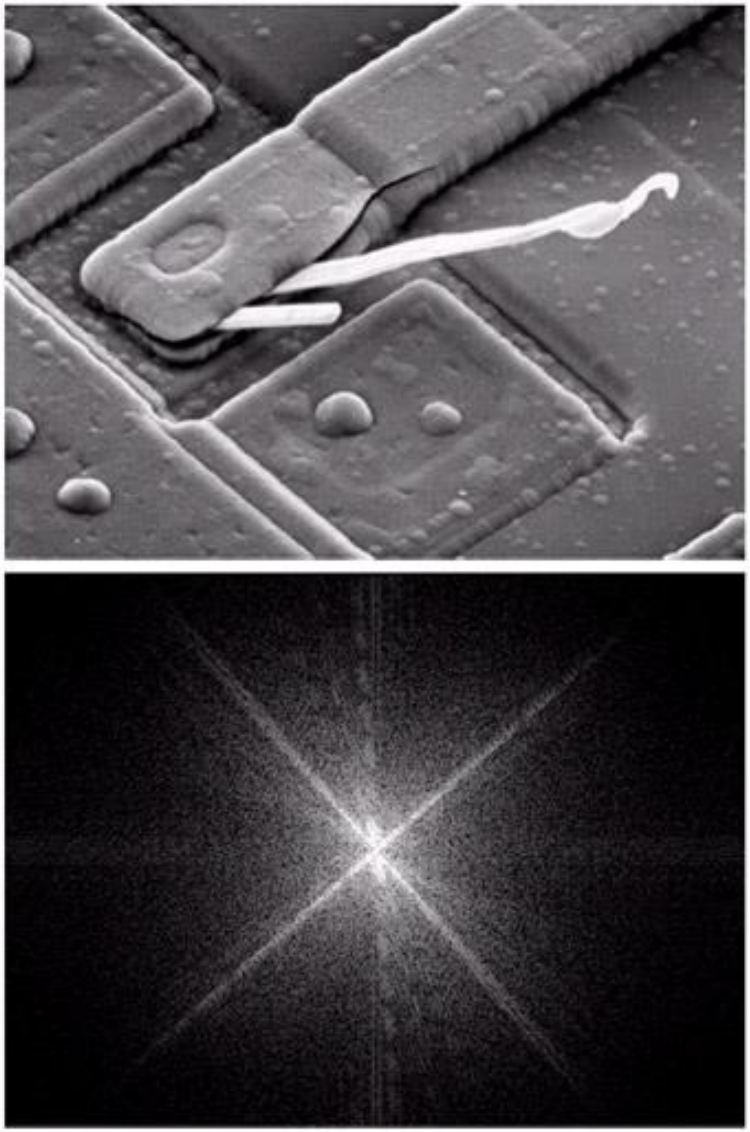


变换能够调整图像的信息熵

变换能够

调整图像

的信息熵



算术编码（Arithmetic Coding）

Let's consider an example of Fixed Arithmetic coding first：

Suppose that there is a 5 symbol sequence $a_1a_2a_3a_3a_4$

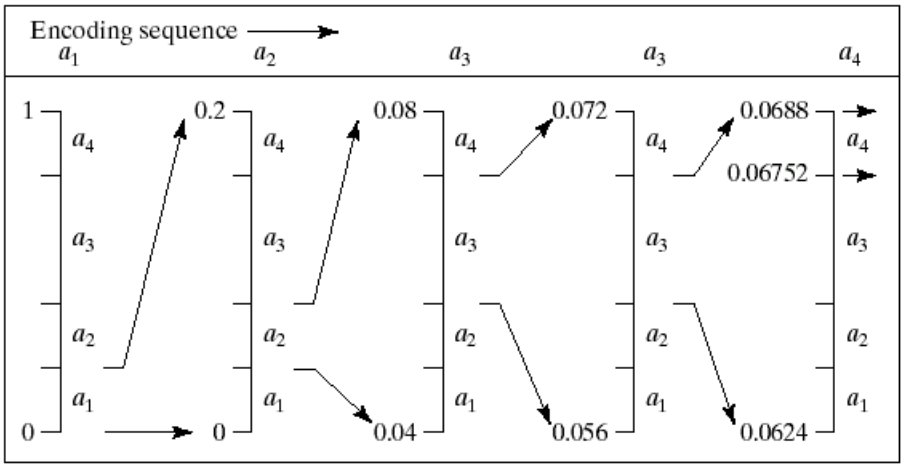


FIGURE 8.13 Arithmetic coding procedure.

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

TABLE 8.6 Arithmetic coding example.

Encoding and Decoding are using the same procedure, but Arithmetic coding has to have ending symbol. Considering this example: a=0.5, b=0.25, c=0.25。character string aacb；aaaa; bbbb; make comparison between Arithmetic coding and Huffman coding.

自适应算术编码（Adaptive Arithmetic Coding）：先预设各字符的概率，然后在编码时再逐个按字符出现的次数和事先给定的规律进行调整。例如，开始时可以将所有的符号设成概率相等，然后在编码过程中根据事先约定好的规则，对各个符号的概率作调整。例如每碰到一次某一个符号，就把这个符号的概率增加0.1，其余符号的概率也根据当前的概率分布作相应的调整。

Error-Free compression

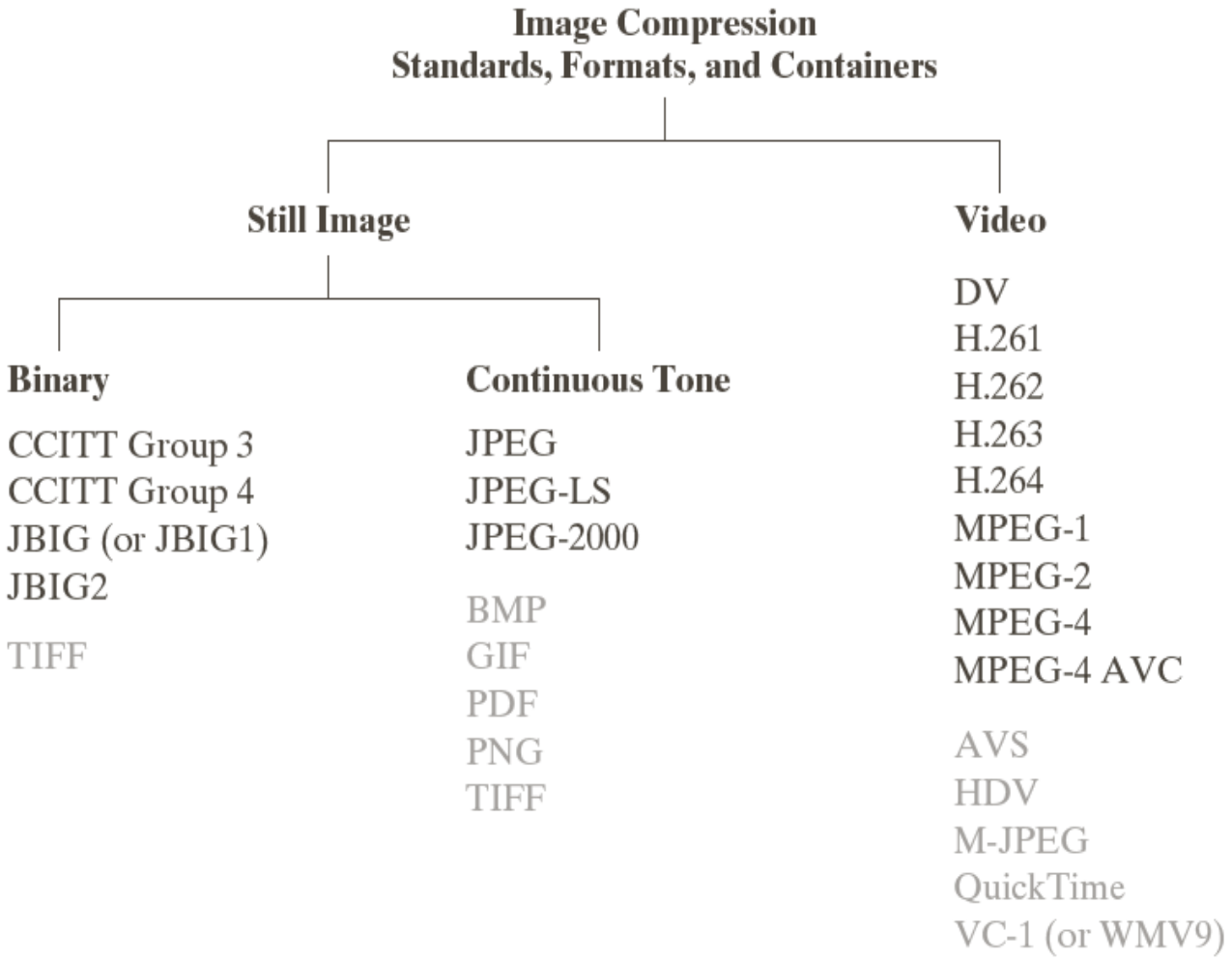
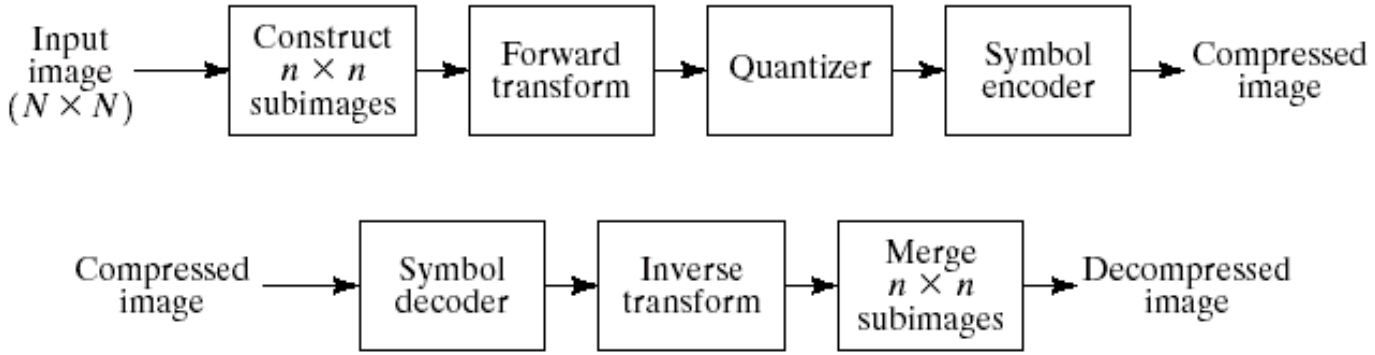


FIGURE 8.6 Some popular image compression standards, file formats, and containers. Internationally sanctioned entries are shown in black; all others are grayed.

有损压缩的一般步骤：



a
b

FIGURE 8.28 A transform coding system: (a) encoder; (b) decoder.

静态图像国际压缩标准JPEG的基本步骤：

图像压缩标准JPEG的基本步骤：

首先将图像分成16x16的小块（宏块），然后对每一宏块做彩色变换，将RGB转换为YCbCr模型， RGB→YCbCr：人眼对亮度更敏感，提取亮度特征，编码时对亮度采用特殊编码。这个变换后能量集中于亮度平面，两个色度平面能量相对少很多，这种格式更有利于提高压缩效率。

$$Y = 0.299R + 0.5870G + 0.1140B$$

$$Cb = -0.1787R - 0.3313G + 0.5000B + 128$$

$$Cr = 0.5000R - 0.4187G - 0.0813B + 128$$

颜色解码：

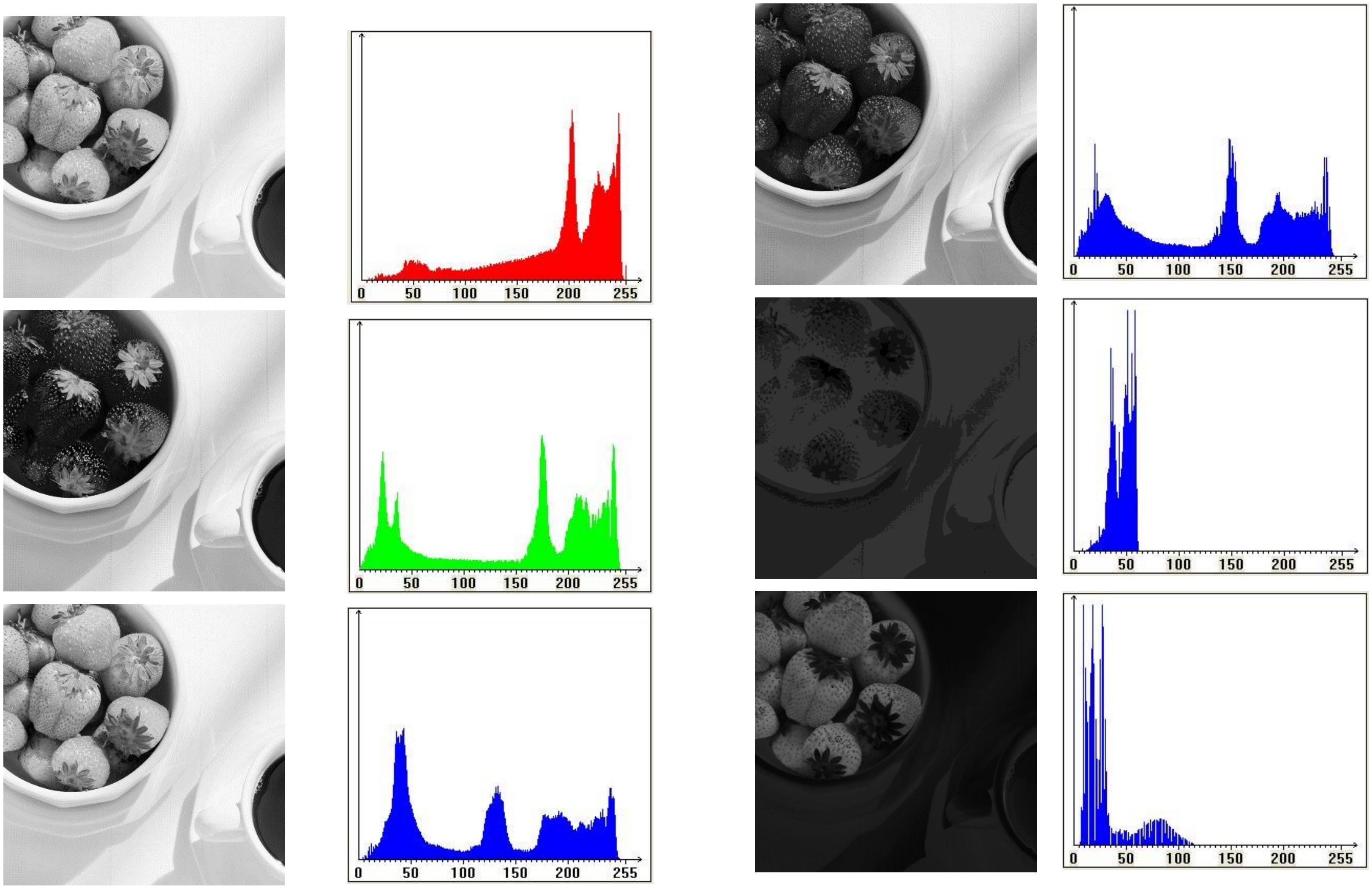
$$R = Y + 1.40200(Cr - 128)$$

$$G = Y - 0.34414(Cb - 128) - 0.71414(Cr - 128)$$

$$B = Y + 1.77200(Cb - 128)$$

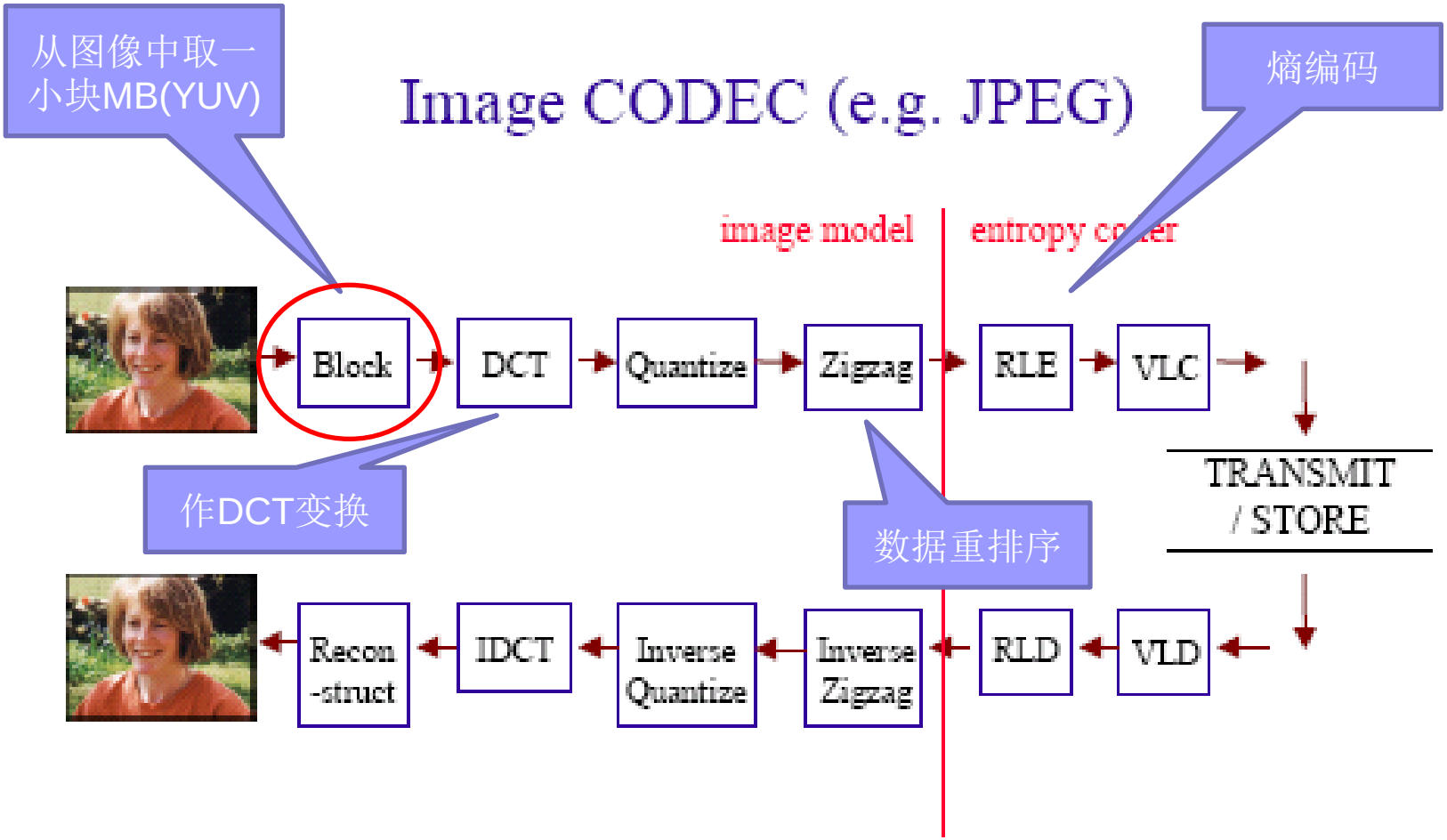
而后将两个色度平面下采样为1/4的像素，这时亮度有4个8x8的小块，色度共有两个，称为4:1:1（有时也称为4:2:0）。

左边两列分别原图的R、G、B颜色分量及对应的直方图，右边两列分别是原图的Y、U、V分量及对应的直方图。各分量的信息熵分别为：R＝7.32，G＝7.49，B＝7.57，Y＝7.57，U＝5.09，V＝5.64，两个色彩空间的平均熵为RGB＝7.46，YUV＝6.10。



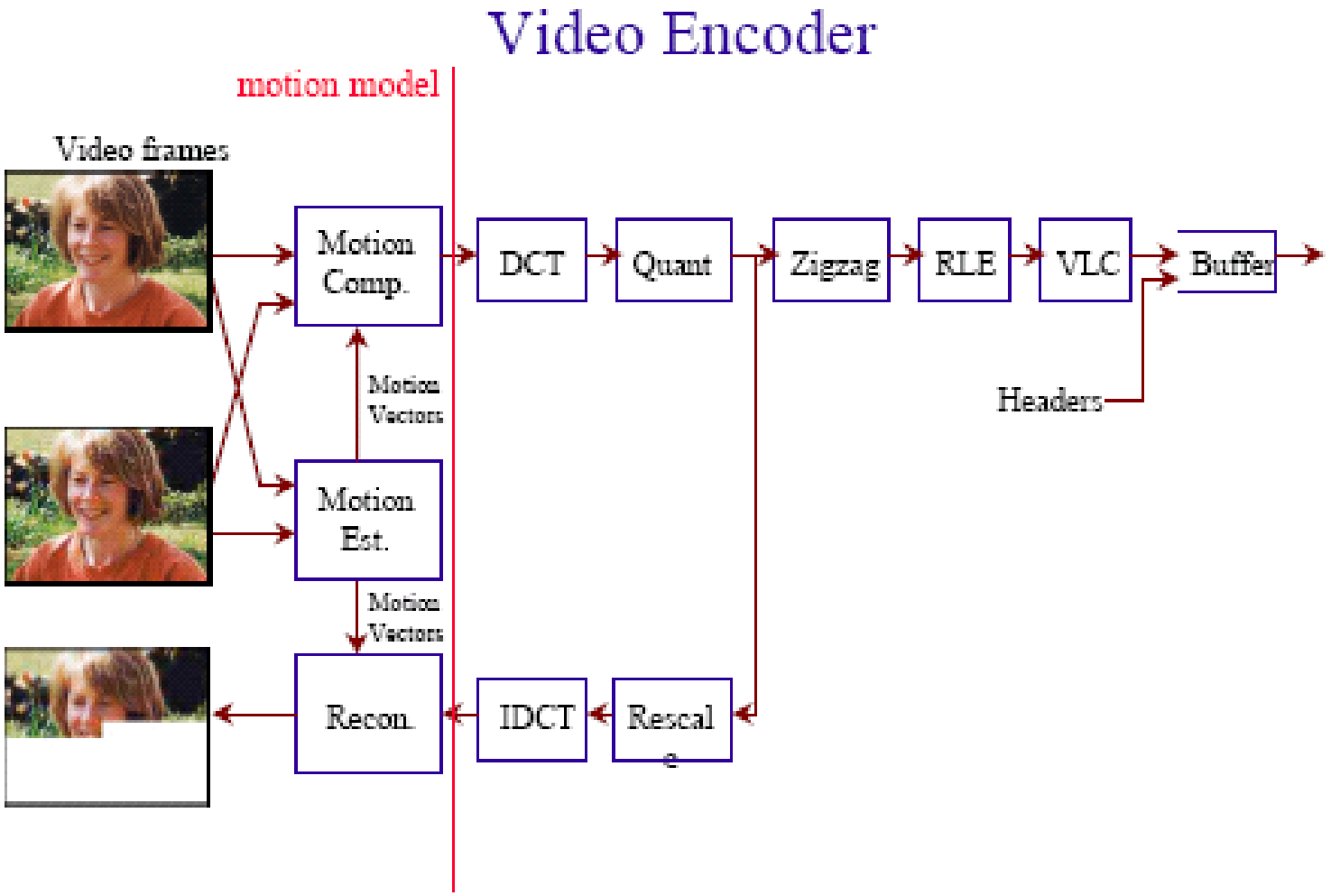
International Compression Standard JPEG

Flow chart：



Which processes in compression will cause this blocky artifact? Explain why.

Natural video encoder



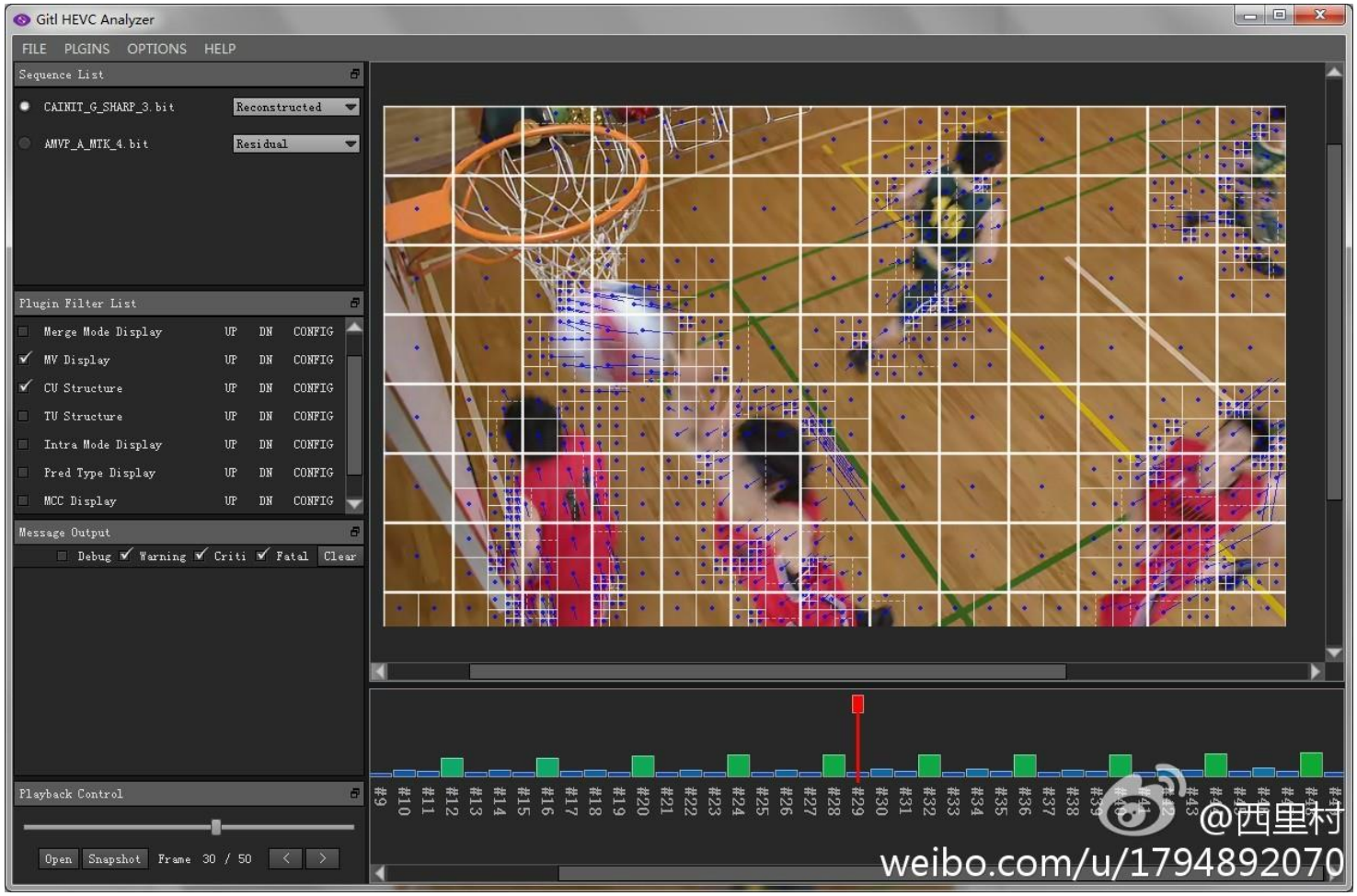
Our open-source software (73 stars)

实验室开源软件（Github）： Gitl HEVC/H.265 Analyzer——H.265码流分析器，供相关研究人员查看HEVC压缩码流编码结构。还可显示CU, TU, MV, Inter, Intra, Bitrate等功能。该分析器还可加载自编的插件对编码结构进行可视化。

已有远远超过1千的下载量，73颗星。目前用户包括名校研究人员和学生、百度、联想以及英国BBC广播电视公司的研发部门（商业用户）等。

https://github.com/lheric/GitlHEVCAnalyzer

Our open-source software (42 stars)



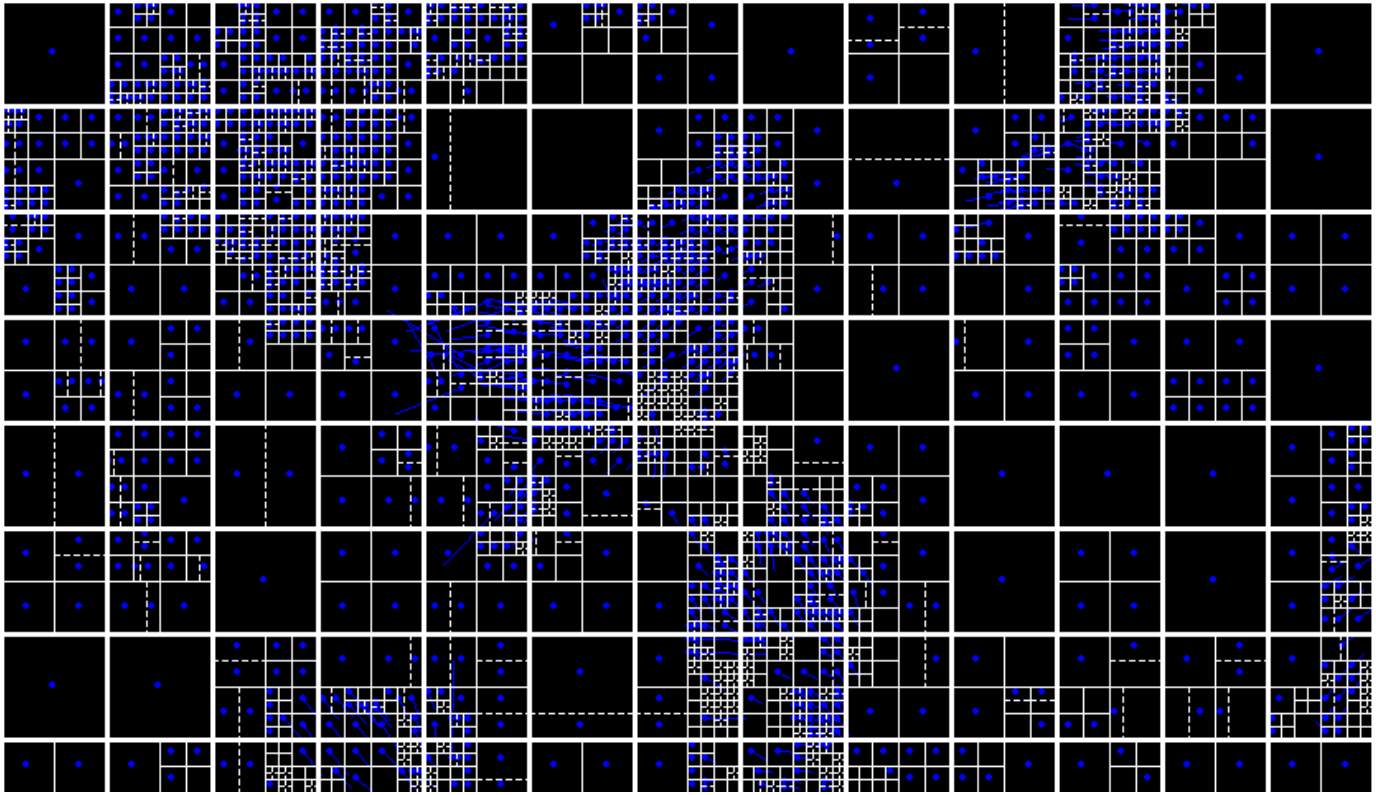
Our open-source software (73 stars)

- **The representation of video data may be divided into two categories: pixel domain and compressed domain.**

Image/video in raw pixel format

Our open-source software (42 stars)

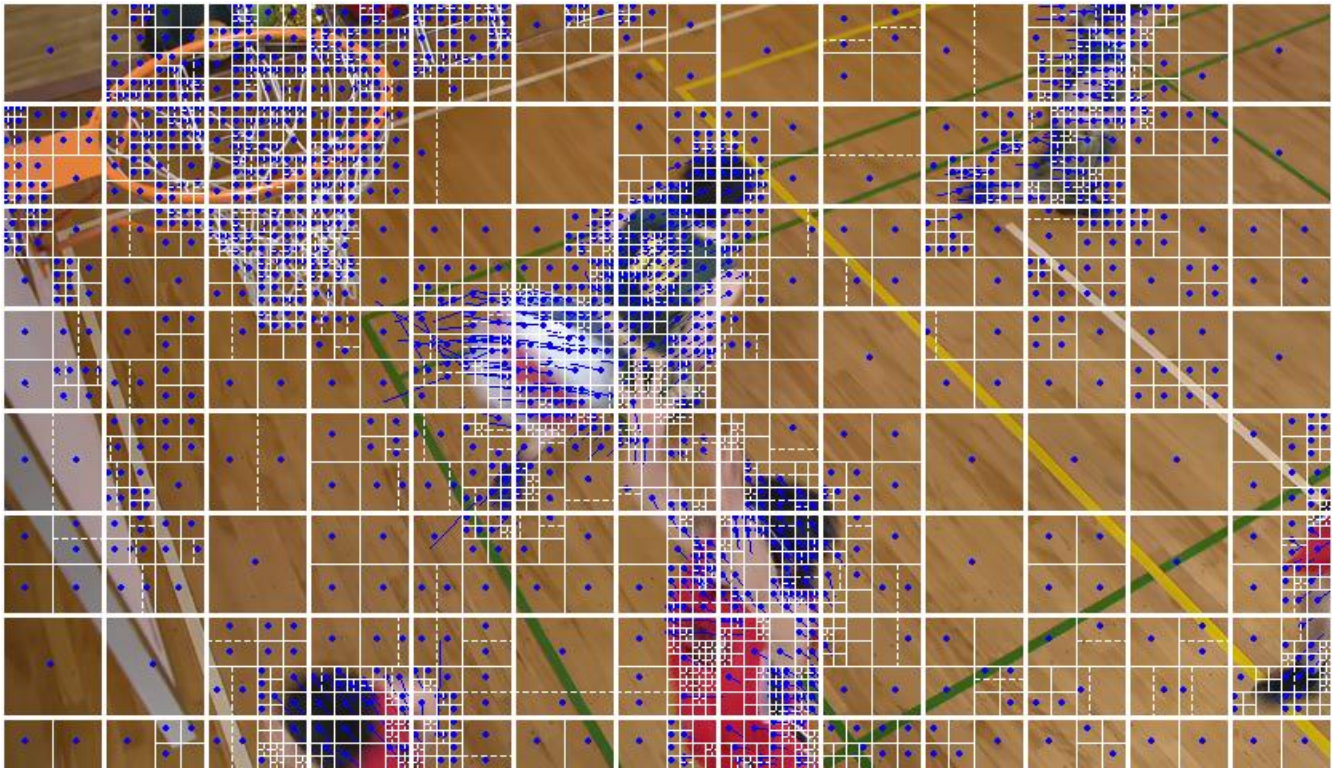
- **The representation of video data** may be divided into two categories: pixel domain and compressed domain.



Image/video in compressed format

Our open-source software (73 stars)

- **The representation of video data** may be divided into two categories: pixel domain and compressed domain.

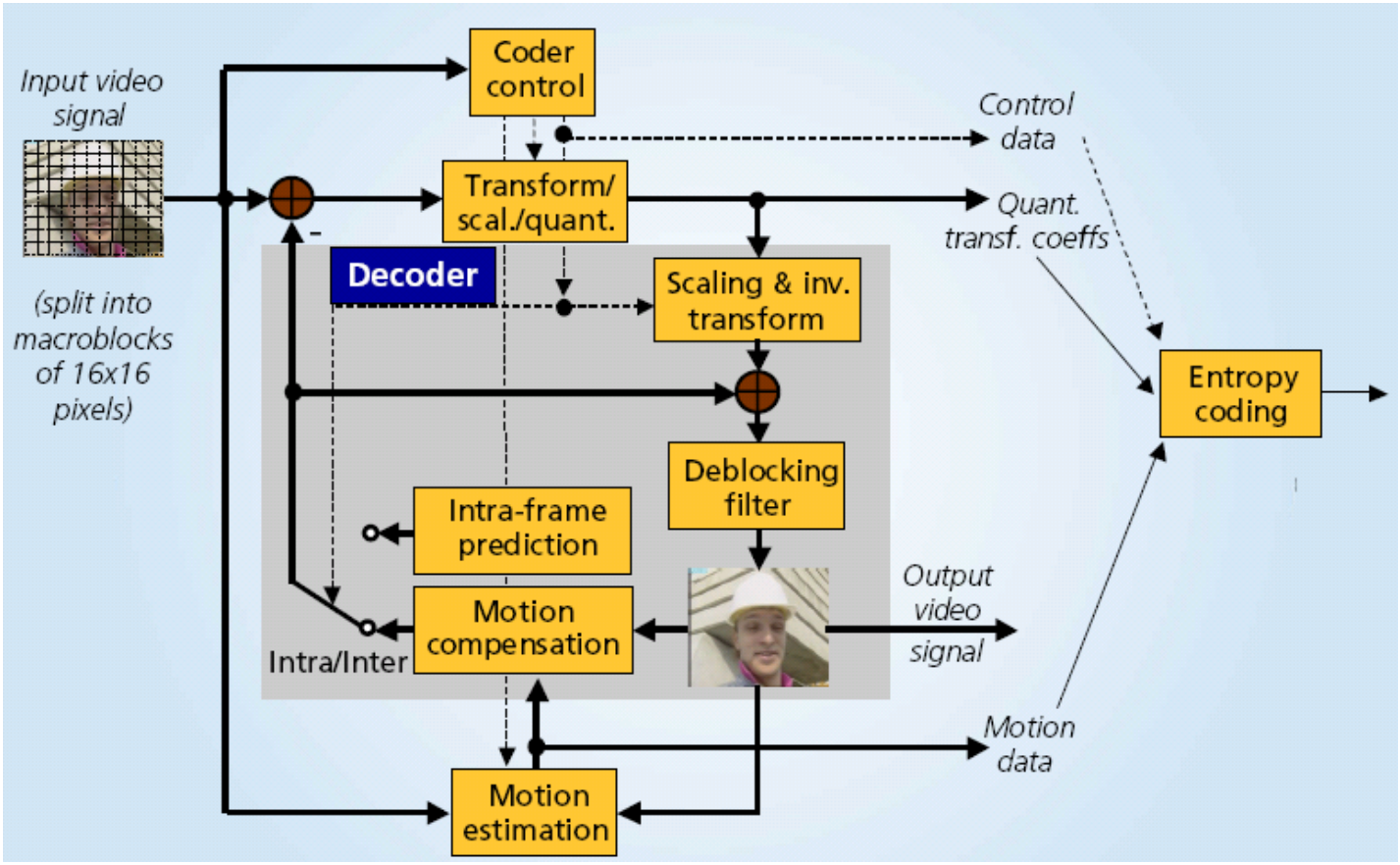


Visualization of compressed format

数码相机工作原理及DIP相关内容

http://blog.sina.com.cn/s/blog_6afbe12601015x3g.html

视频编码总体框图



视频编码技术是在现有图像编码技术的基础上加入了提取时间冗余度的技术（帧间技术）。其中，对于视频中关键帧的编码就相当于静态图像编码，而对其余图像帧的编码则是先利用已编码帧的图像对当前图像预测，然后再对其残差信号进行编码。

视频编码概述

视频编码技术是在现有图像编码技术的基础上加入了提取时间冗余度的技术（帧间技术）。其中，对于视频中关键帧的编码就相当于静态图像编码，而对其余图像帧的编码则是先利用已编码帧的图像对当前图像（或MB）预测，然后再对其残差信号进行编码。

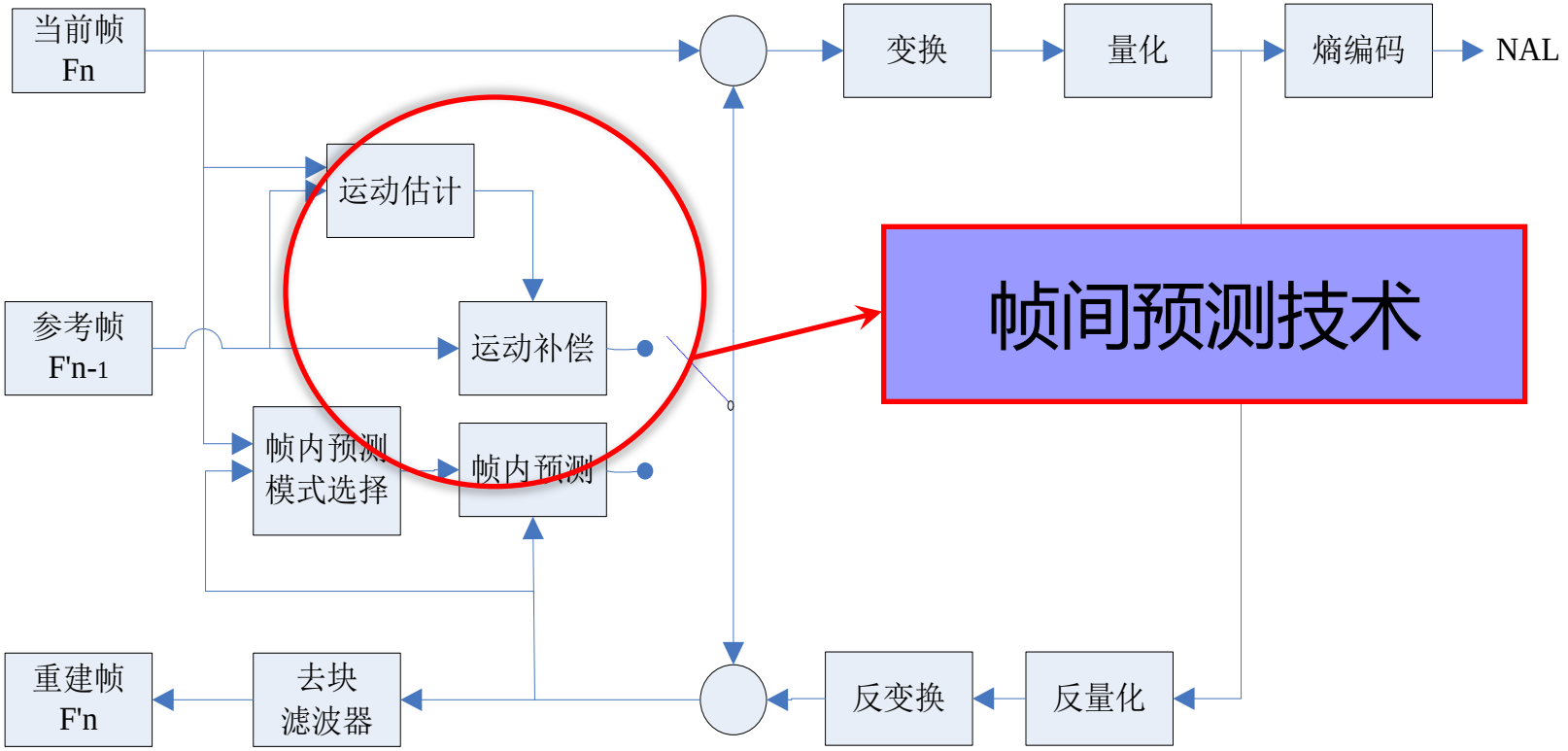
我们目前的研究工作主要针对可选技术中的帧间预测技术进行计算复杂度和率失真的优化。

视频编码技术分类

- 一类是必选技术，如变换、量化、熵编码等。这类技术是视频编码标准码流的基本要求。
- 另一类为可选技术，如帧内预测、帧间预测（包括整数、分数运动估计、多模式、多参考帧等）、环路滤波等。可选技术的采用与否，不影响视频编码所产生码流的兼容性，但却对视频的编码效率和率失真性能以及计算资源的消耗有着重要的影响。

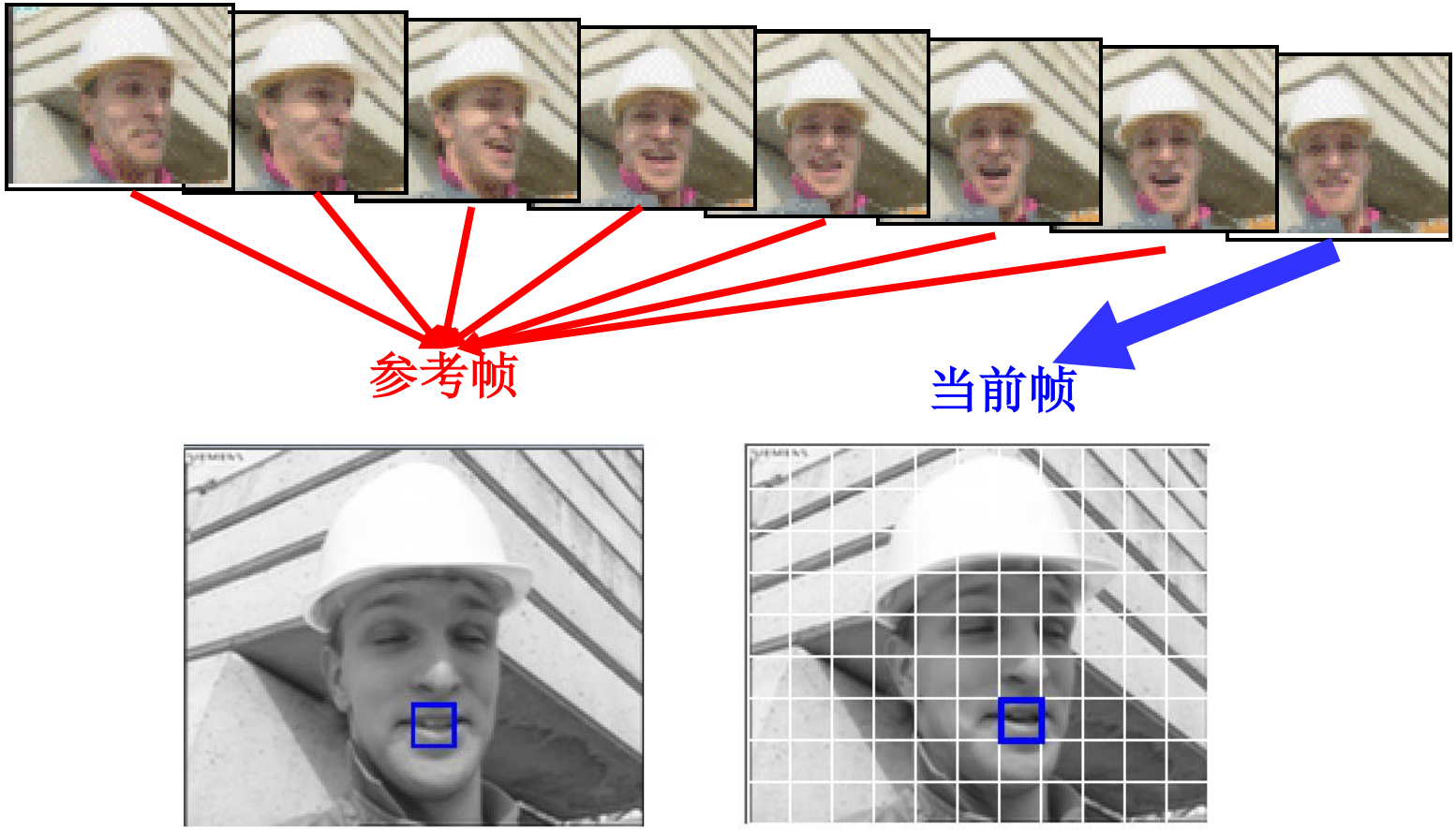
我们主要针对可选技术中的帧间预测技术进行计算复杂度和率失真的优化。

什么是帧间预测技术



帧间预测是视频编码中用以减少时间冗余度的技术，利用已编码帧的图像对当前图像预测。

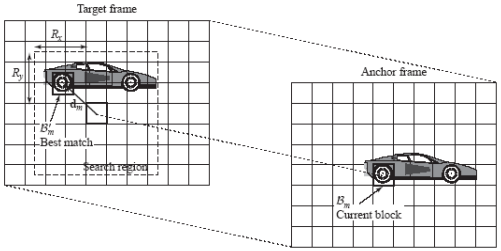
帧间预测技术

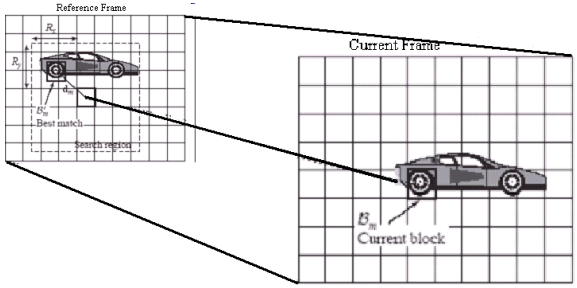


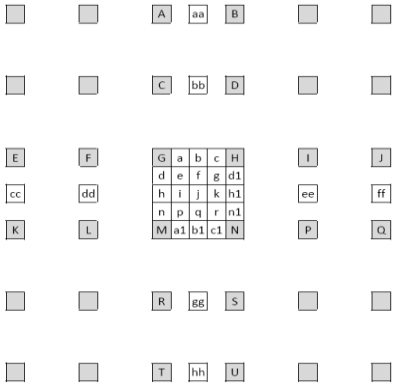
帧间预测技术包括：整/分数运动估计，分数点插值，多模式决策，多参考帧选择等。

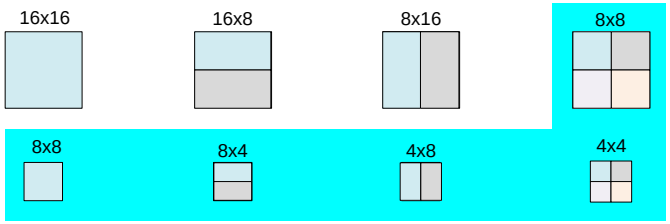
视频编码标准H.264的帧间技术

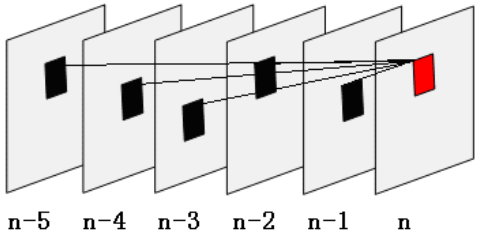
- 整数运动估计
- 分数运动估计
- 分数插值
- 多模式决策
- 多参考帧选择



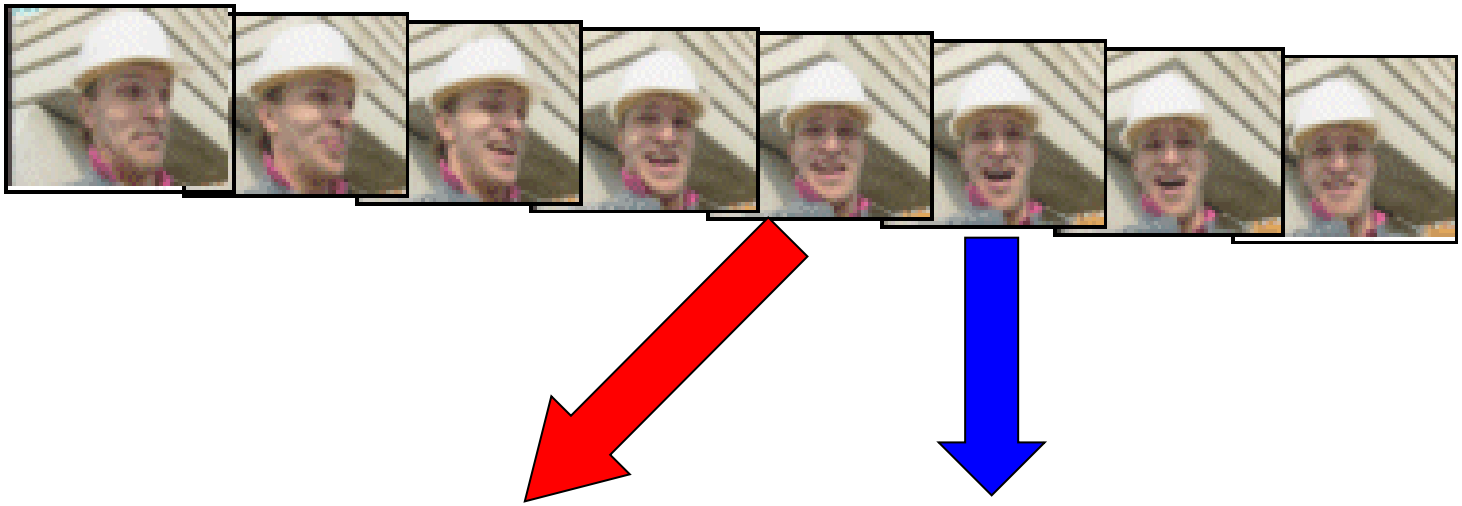






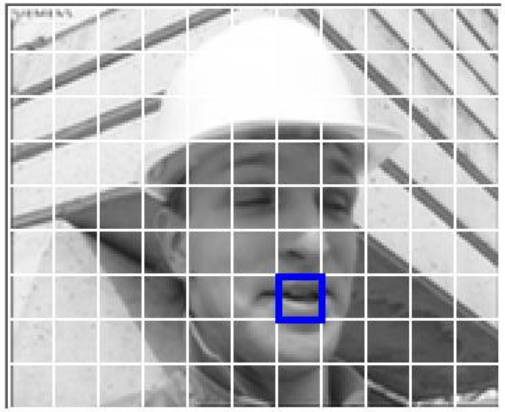


分数运动估计的先驱：整数运动估计



参考帧

当前帧



¼ 像素分数运动估计和分数插值

当前帧

参考帧

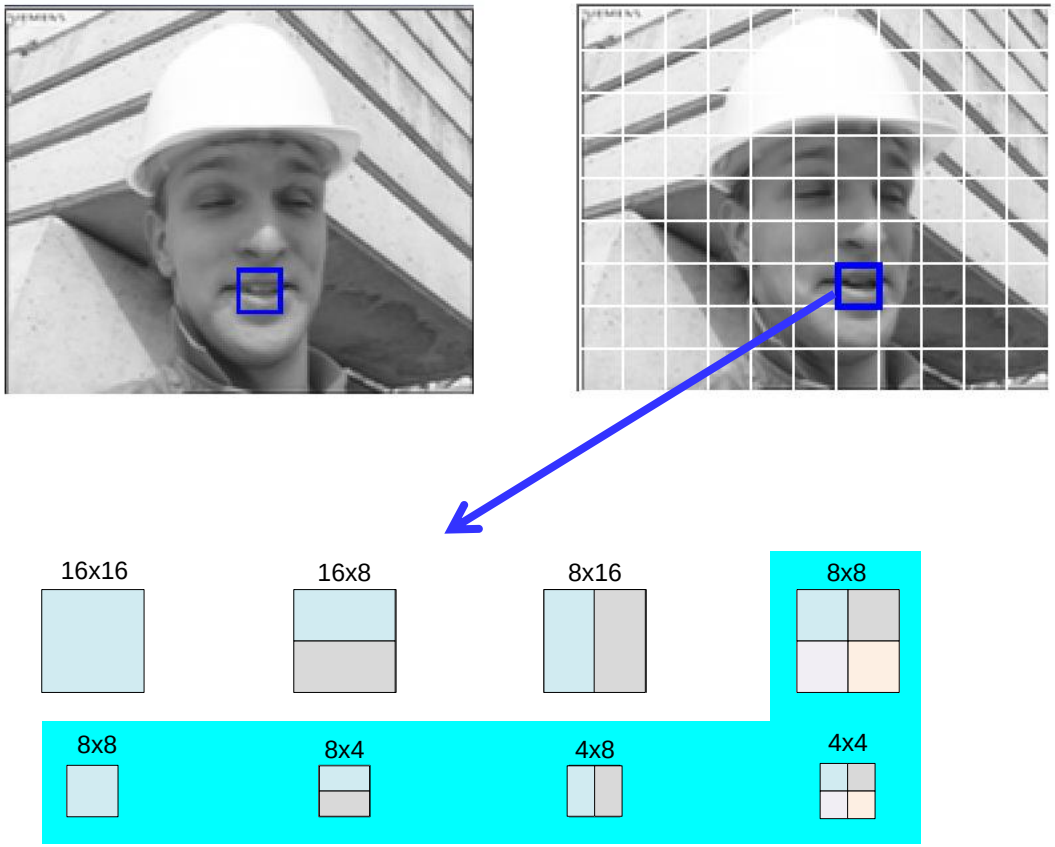
	0	¼	½	¾
0				
¼				
½				
¾				

分数运动估计和插值提高了编码的率失真新能，
但占用了非常大部分的计算时间（Minoor'07 TCSVT）。

H.264 多模式决策

参考帧

当前帧



研究思路

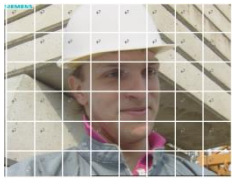
- 在序列层面：

  - 研究不同帧间技术作用在每帧中的性价比优先顺序。

- 在帧层面：

  - 研究各种帧间技术作用视频不同部分（即不同宏块）的性价比优先顺序。

- 在宏块层面：



  - 为不同的帧间技术设计对应的算法，并使这些算法每一步都获得最大的率失真收益和消耗最少的计算资源，随时停止时都可得到最优的性能。

目前已有的研究工作

- 在宏块层面：
  - 基于性价比优先原则的分数运动估计和插值
  - 扩展宏块尺寸（EMS）的多模式决策
- 在帧层面：
  - 计算能力可伸缩的整数运动估计率失真优化
  - 计算复杂度可变的多模式决策

下一步的研究工作

- 近期：
  - 把现有的结果推广到新的编码标准HECV
- 长期：
  - 继续深化相关研究
  - 提交自己的编码标准提案？？？