

Chapter 1: Media and Systems

一. 多媒体: 有多种方式/中间介质来传递信息(图像, 声音, 触觉等)

不同的媒体环境: 社交媒体: facebook, twitter, instagram

大众传媒: 报纸, 杂志, 电视节目

平台: 移动电话, 电脑, 电视, 纸媒

二. 数字多媒体:

多媒体: 跨学科, 面向应用的技术; 利用人类的感知 & 计算机存储的功能, 控制, 传递信息.

数字多媒体: 关注文本集成, 图表, 图像等可以数字化地展现, 存储, 传输及处理的领域

多媒体属性: ① 数字化: 媒介以数字形式展现

② 计算机化: 由计算机处理和控制

③ 分布式: 信息从远端中端进行中继 (提前存储或实时产生的信息)

④ 相互作用: 可根据用户需求查找, 浏览, 过滤, 检索及展示媒体.

⑤ 集成性: 以集成的形式呈现出来

多媒体考虑: ① 重现: 采样, 量化, 编码

② 存储: 大的存储需求及新的接入模式

Chapter 2: 图像及视频压缩

1. 发展: 香农提出熵: 度量信源信息量

解决 2 个问题: ① 如何对信息进行有效传输 —— 信源编码 (压缩信号)

② 如何使信息在噪声存在的信道中可靠传输 —— 信道编码 (一种差错控制编码)

现代多媒体技术: 依赖于压缩算法实现.

① 减少了多媒体信息所需存储容量.

② 减小了信息传输需要的比特率.

① 编码冗余 {
 空间冗余: 框内冗余, 指图像的像素点之间的相关性.
 时间冗余: 框间冗余, 指连续帧的像素点之间的相关性.
 编码冗余: 关注编码本身, 分为均匀长度编码及变长编码

② 视觉冗余 {
 频率: 视觉对高频、对角线中噪声不敏感
 颜色: 视觉对图像亮度比颜色敏感

2. 图像及视频压缩系统:

① 信源编码: 由原始信号的内容决定, 也称基于语义的编码; 编码形式: 转换编码; 差分编码; 矢量量化

与严格熵编码相比, 有更高压缩率. 并在无损/有损模式下都可以运行.

无损压缩: 重构的解压图像与原始图像一致, 用于重要数据 (医学)

有损压缩: 重构的解压图像与原始图像不同; 部分数据可以被丢弃 (人眼无法区分)

input → [图片/视频信号压缩] → [传输/存储] → [信号重现] → 输出

② 转换编码: 将数据转换为更容易被压缩的形式. 为压缩提供有利条件:

① 减少了无关数据, 使得转换系数更容易被量化.

② 减少了转换系数间的相关性, 使得冗余降低.

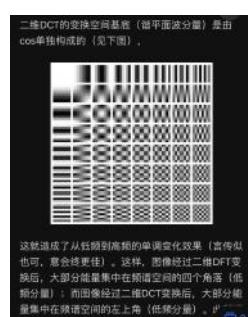
③ 转换提供了能量压缩.

逆转换后可以得到原始信号.

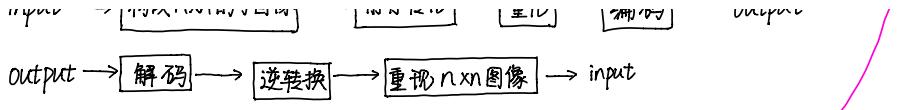
基于转换的数据压缩含: 转换器, 量化器, 编码器

原图转化为更适合传输的感

仅为更适应传输



input → [输出 nxn 的子图像] → [前向转换] → [量化] → [逆 DCT] → output



① 转换器：实现图像一对一的转换；转换后图像更适合压缩；实现能量压缩（DCT 转化后能量集中于低频，集中在左上角）
常用 DFT、DCT、DWT、KLT

② 量形器：转换系数中选取有限个符号；多对一的映射会丢失信息，且是不可逆的。

分为标量量化：数据逐个进行量化

矢量量化：数据按数据块进行量化

③ 编码器：为每个符号分配码字或比特流。（可使用定长/变长编码）

变长编码(VLC) & 熵编码为每个符号（量化后整序）分配最小长度二进制码元。

转换：基本函数的变换，图像视角/方向的变换

④ DCT (Discrete cosine transform)

$$S_{ij} \xrightarrow{2D-DCT} S_{uv}$$

$$S_{uv} = \alpha(u) \cdot \alpha(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} S_{ij} \cdot \cos\left(\frac{(2i+1)u\pi}{2N}\right) \cdot \cos\left(\frac{(2j+1)v\pi}{2N}\right)$$

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & k=0 \\ \sqrt{\frac{2}{N}} & k=1, 2, \dots, N-1 \end{cases}$$

使用 DCT 使变换系数之间冗余减少且能量压缩。

优点：转换函数是固定的，与图像无关；缺点：压缩不如别的转换一样有效。

⑤ KLT (Karhunen-Loeve transform)：用于图像压缩的重要方法，通过旋转坐标轴展示数据子空间。也称 PCA、EDA

优点：有利于能量压缩，系数互相关，降维

缺点：①不同图像需要不同基础函数 ②需估计其协方差矩阵 ③计算量大

过程：① 将图像分割为 $N \times N$ 图像块

② 将每个 $N \times N$ 图像块用 X_{ni} 表示 \Rightarrow 形成了一个 $N \times N$ 的矩阵。

③ 将图像转换为以图像均值为中心的图像： $X = X_{ni} - \bar{X}$

④ 计算协方差矩阵： $C = E[X X^T] = E[(X_{ni} - \bar{X})(X_{ni} - \bar{X})^T]$

$$\Rightarrow C = \frac{1}{N-1} \sum_{i=1}^{N-1} (X_{ni} - \bar{X})(X_{ni} - \bar{X})^T \quad (\text{去除了中心})$$

$$\bar{X}_n = \frac{1}{N-1} \sum_{i=1}^{N-1} X_{ni} \quad (\text{所有的均值})$$

⑤ 协方差矩阵各分量： $C P_i = \lambda_i P_i$ $i=1, 2, 3, \dots, M$ ，其中 $M=N^2$

⑥ KLT 的基本函数是协方差矩阵 C 的特征向量。

⑦ 决定基本函数取值：特征向量 特征值大。

$$\Rightarrow \vec{x} = \sum_{i=1}^M C_i \vec{P}_i \Rightarrow C_i = \vec{P}_i^T \vec{x}$$

$$E[C_i C_j] = \begin{cases} \lambda_i & i=j \\ 0 & i \neq j \end{cases}$$

● low-rank 模型：

① 假设特征值都不同且降序排列： $\lambda_1 > \lambda_2 > \dots > \lambda_M$

② 利用 low-rank 模型将数据向量近似为 $\hat{x} = \sum_{i=1}^r C_i \vec{P}_i$ ($r < M$)

③ 差分/预测编码：适用于采样相似信号

在图像领域很小时，像素值相似 实际上，差别很小 ??

对像素有区别的部分进行编码比对像素编码更有效。

编码① 预测器利用相邻像素预测当前像素实际值，并取最相近整数。

② 计算并编码 预测误差，输出以比特流形式被存储。

解码：解码比特流获得预测误差

⇒ 用解码后图像相邻像素预测当前像素。

⇒ 预测误差 + 图像的预测值重构像素值。

⇒ 由一系列像素值重构图片

④ 矢量量化：在小的图像块中像素多为相关的；用常见的图像块及其索引（码字或码向量）表示图像（不为各像素编码）；码字的合集 \Rightarrow 索码本
编码：① 图像被分为方块 \Rightarrow 称为向量（字典中的顺序）

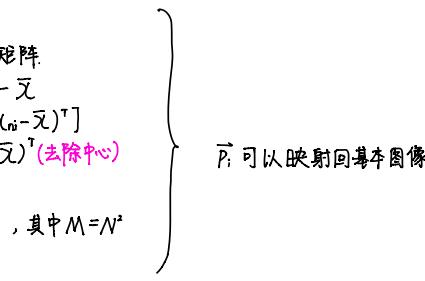
② 索码本：寻找各图像块最匹配的码字（图像模式）

③ 将码字的索引以压缩比特流进行存储。

④ 编/解码器都有索码本，故仅存储索引。

⑤ 若无相同码字则使用最相近码字，解码时会失真。

解码：① 从压缩比特流中提取索引



⑤ 恢复重建

- ① JPEG根据图像/帧建立比特流规则
- ② 恢复重建器封装与编码图像相关的信息

帧 → 图像 → 图像块 → 图像块 → DC变换后为ID格式(skip, value)

③ JPEG解码器(逆过程)

JPEG解码器提取控制信息及表, 传统图像生成器

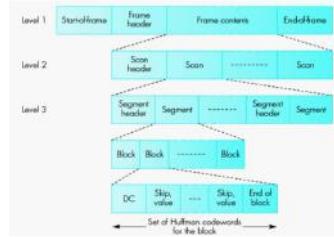
⇒ 哈夫曼解码: 获取DC及AC符号

⇒ 行程长度压缩解码: 获得AC系数; 差分解码: 获得DC系数

⇒ DC及AC系数合并在一起并使用量化表去量化

⇒ DCT逆变换将DCT系数转换为像素块

⇒ 图像构建器将像素块转化为图像



⑥ 基于顺序DCT的JPEG

在基于顺序DCT的模式中, 图像被分割为 8×8 像素的图像块并从左到右, 从上到下被处理

将2D-DCT应用于每个 8×8 像素块, 并量化DCT系数

量化的DCT系数进行熵编码, 以压缩比特流输出

⑦ 基于渐近DCT的JPEG

基于渐近DCT的编码需通过多次扫描

每次扫描中, 量化的DCT系数, 利用频谱选择或连贯进似的方式编码

频谱选择: 将系数分为多个频段, 每次扫描仅编码单个频段系数

连贯进似: 对系数中数量更多的编码, 再对较少的编码

⑧ 无损模式JPEG: 基于空间的编码

各像素使用预测编码, 预测值由ID或2D预测器生成

⑨ 分层模式JPEG

a. 将图像分为一系列帧, 每帧通过下采样产生(过了低通滤波器)

b. 第一帧利用非差分熵编码, 其余帧以差分形式被编码