

- 第三十二讲：基变换和图像压缩
  - 图像压缩
    - 傅里叶基
    - 小波基
  - 基变换

## 第三十二讲：基变换和图像压缩

### 图像压缩

本讲我们介绍一种图片有损压缩的一种方法：JPEG。

假设我们有一张图片，长宽皆为512个像素，我们用 $x_i$ 来表示第 $i$ 个像素，如果是灰度照片，通常 $x_i$ 可以在 $[0, 255]$ 上取值，也就是8 bits。对于这承载这张图片信息的向量 $x$ 来说，有 $x \in \mathbb{R}^n, n = 512^2$ 。而如果是彩色照片，通常需要三个量来表示一个像素，则向量长度也会变为现在的三倍。

如此大的数据不经过压缩很难广泛传播。教学录像采用的压缩方法就是JPEG（Joint Photographic Expert Group，联合图像专家组），该方法采用的就是基变换的方式压缩图像。比如说一块干净的黑白，其附近的像素值应该非常接近，此时如果一个像素一个像素的描述黑白灰度值就太浪费空间了，所以标准基在这种情况下并不能很好的利用图片的特性。

我们知道，标准基是  $\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \cdots \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$ ， 我们想寻找一个更好的基。

我们试试使用别的基描述图片，比如：

- 基中含有的一个向量  $[1 \ 1 \ \cdots \ 1]^T$ ，即分量全为1的向量，一个向量就可以完整的给出所有“像素一致图像”的信息；
- 另一个向量  $[1 \ -1 \ \cdots \ 1 \ -1]^T$ ，正负交替出现，比如描述国际象棋棋盘；
- 第三个个向量  $[1 \ 1 \ \cdots \ -1 \ -1]^T$ ，一半正一半负，比如描述一半亮一半暗的图片；

# 傅里叶基

现在我们来介绍傅里叶基，以 $8 \times 8$ 傅里叶基为例（这表示我们每次只处理 $8 \times 8$ 像素的一小块图像）：

$$F_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{n-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{n-1} & w^{2(n-1)} & \cdots & w^{(n-1)^2} \end{bmatrix}, w = e^{i2\pi/n}, n = 8, \text{ 我们不需要深入8阶}$$

傅里叶基的细节，先看看使用傅里叶基的思路是怎样的。

每次处理 $8 \times 8$ 的一小块时，会遇到64个像素，也就是64个基向量，64个系数，在64维空间中利用傅里叶向量做基变换：

- 输入信号 $x$ 为64维向量 $\xrightarrow{\text{基变换}}$ 输出信号 $c$ 为 $x$ 在傅里叶基下的64个系数。

注意前面做的都是无损的步骤，我们只是选了 $\mathbb{R}^{64}$ 的一组基，接着把信号用这组基表达出来。

接下来的步骤就涉及到压缩和损失了：

- 一种方法是扔掉较小的系数，这叫做阈值量化（**thresholding**），我们设定一个阈值，任何不在阈值范围内的基向量、系数都将丢弃，虽然有信息损失，但是只要阈值设置合理，肉眼几乎无法区别压缩前后的图片。经由此步处理，向量 $c$ 变为 $\hat{c}$ ，而 $\hat{c}$ 将有很多0。

通常 $[1 \ 1 \ \cdots \ 1]^T$ 向量很难被丢弃，它通常具有较大的系数。但是 $[1 \ -1 \ \cdots \ 1 \ -1]^T$ 向量在平滑信号中的可能性就很小了。前一个的向量称作低频信号，频率为0，后一个向量称作高频信号，也是我们能够得到的最高频率的信号，如果是噪音或抖动输出的就是它。

比如讲课的视频图像信号，这种平滑的情形下输出的大多是低频信号，很少出现噪音。

- 接着我们用这些系数 $\hat{c}$ 来重构信号，用这些系数乘以对应的基向量 $\hat{x} = \sum \hat{c}_i v_i$ ，但是这个求和不再是64项求和了，因为压缩后的系数中有很多零存在，比如说我们压缩后 $\hat{c}$ 中仅有三个非零项，那么压缩比将近达到21 : 1。

我们再来提一下视频压缩：视频是一系列连续图像，且相近的帧非常接近，而我们的压缩算法就需要利用这个相近性质。在实际生活中，从时间与空间的角度讲，事物不会瞬间改变。

## 小波基

接下来介绍另一组基，它是傅里叶基的竞争对手，名为小波（**wavelets**），同样以 $8 \times 8$

为例：

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}。$$

可以看出傅里叶基中频率最高的向量为小波后四个基向量之和。

在标准基下的一组（按八个一组计算， $P \in \mathbb{R}^8$ ）像素 $P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_8 \end{bmatrix} = c_1 w_1 + c_2 w_2 + \dots + c_n w_n = [w_1 \ w_2 \ \dots \ w_n] \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$ ，即 $P = WC$ ，我们需要计算像素向量在另一组基下系数，所以有 $C = W^{-1}P$ 。

此时我们发现，如果选取“好的基”会使得逆矩阵的求解过程变简单，所谓“好的基”：

- 计算快；

我们需要大量使用 $P = WC$ 来计算整幅图在另一个基下的表达，在傅里叶变换中我们学习了快速傅里叶变换（FFT），同样的在小波变换中也有快速小波变换（FWT）；

另外的，我们需要计算其逆矩阵，所以这个逆矩阵计算也必须快，观察小波基不难发现基向量相互正交，假设我们已经对小波基做了标准化处理，则小波基是一组标准正交基，所以有 $W^{-1} = W^T$ 。

- 仅需少量向量即可最大限度的重现图像。

因为在图像压缩时，我们会舍弃较小的系数，比如 $c_5, c_6, c_7, c_8$ ，所以后四个的基向量都会被舍弃，重现图像时仅使用前四个基向量的线性组合，而好的基选取会在使用较少基的前提下保证图像质量不会有较大损失。

题外话：JPEG2000标准会将小波基纳入压缩算法。我们上面介绍的是最简单的一组小波基，而FBI的指纹识别或JPEG2000的压缩算法纳入的是更加平滑的小波基，不会使用像上面介绍的那种直接从1变为-1的基。

要想继续了解小波基，可以参考一篇非常精彩的文章[能不能通俗的讲解下傅立叶分析和小波分析之间的关系？——“咚懂咚懂咚”的答案](#)

## 基变换

---

前面介绍小波基的时候我们就已经做了一次基变换。

将目标基的向量按列组成矩阵 $W$ ，则基变换就是 $[x] \xrightarrow{x=Wc} [c]$ 。

看一个例子，有线性变换 $T: \mathbb{R}^8 \rightarrow \mathbb{R}^8$ ，在第一组基 $v_1, v_2, \dots, v_8$ 上计算得到矩阵 $A$ ，在第二组基 $w_1, w_2, \dots, w_n$ 上计算得到矩阵 $B$ 。先说结论，矩阵 $A, B$ 是相似的，也就是有 $B = M^{-1}AM$ ，而 $M$ 就是基变换矩阵。

进行基变换时会发生两件事：

1. 每个向量都会有一组新的坐标，而 $x = Wc$ 就是新旧坐标的关系；
2. 每个线性变换都会有一个新的矩阵，而 $B = M^{-1}AM$ 就是新旧矩阵的关系。

再来看什么是 $A$ 矩阵？

对于第一组基 $v_1, v_2, \dots, v_8$ ，要完全了解线性变换 $T$ ，只需要知道 $T$ 作用在基的每一个向量上会产生什么结果即可。因为在这个基下的每一个向量都可以写成 $x = c_1v_1 + c_2v_2 + \dots + c_8v_8$ 的形式，所以 $T(x) = c_1T(v_1) + c_2T(v_2) + \dots + c_8T(v_8)$ 。

而且 $T(v_1) = a_{11}v_1 + a_{21}v_2 + \cdots + a_{81}v_8$ ,  $T(v_2) = a_{12}v_1 + a_{22}v_2 + \cdots +$

$$a_{82}v_8, \cdots, \text{ 则矩阵 } [A] = \left[ \begin{array}{c|c|c|c} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array} \right]$$

这些都是上一讲结尾所涉及的知识。

最后我们以一个更加特殊的基收场，设 $v_1, v_2, \cdots, v_n$ 是一组特征向量，也就是 $T(v_i) = \lambda_i v_i$ ，那么问题就是矩阵 $A$ 是什么？

继续使用线性变换中学到的，输入的第一个向量 $v_1$ 经由 $T$ 加工后得到 $\lambda_1 v_1$ ，第二个向量 $v_2 \xrightarrow{T} \lambda_2 v_2$ ，继续做下去，最终有 $v_n \xrightarrow{T} \lambda_n v_n$ 。除了 $\lambda_i v_i$ 外的其他基向量都变为

$$0, \text{ 那么矩阵 } A = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}。$$

这是一个非常完美的基，我们在图像处理中最想要的就是这种基，但是找出像素矩阵的特征向量代价太大，所以我们找了一些代价小同时效果也不错的基，比如小波基、傅里叶基等等。