

图像变换是将图像信号从空域变换到其他域进行分析,常用的图像变换包括图像频域变换(傅里叶变换)、离散余弦变换、Gabor变换和小波变换等。本章就这几种典型的图像变换进行分析和介绍,最后给出基于主成分分析、字典学习等前沿的图像变换算法。

3.1 傅里叶变换

3.1.1 傅里叶级数

傅里叶级数:对于周期为 T 的周期信号 $f(x)$ (即 $f(x) = f(x + mT)$, m 为任意整数),其基角频率为 $\omega = 2\pi/T$,则该周期信号可以分解为在 $(-T/2, T/2)$ 上的正交函数 $\sin(x\omega n)$ 与 $\cos(x\omega n)$ (n 为任意整数)的线性组合,即

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{+\infty} a_n \times \cos(x\omega n) + b_n \times \sin(x\omega n) \quad (3-1-1)$$

其中,

$$\begin{cases} a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(x) \cos(x\omega n) dx, & n = 0, 1, 2, \dots \\ b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(x) \sin(x\omega n) dx, & n = 0, 1, 2, \dots \end{cases} \quad (3-1-2)$$

非周期信号没有傅里叶级数,但是可以把非周期的信号看成 $T = \infty$ 的周期信号,用积分的思想仍然可以把任意函数表示成正弦波和余弦波的叠加。对于信号长度为 N 的非周期信号 $f(x)$,若将其周期 T 看作 N ,在有限正弦波和余弦波叠加下,则该信号傅里叶级数表示为

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^P a_n \times \cos(x\omega n) + b_n \times \sin(x\omega n), \quad \omega = \frac{2\pi}{N} \quad (3-1-3)$$

其中, P 表示正弦波和余弦波的个数。根据欧拉公式,将式(3-1-3)改写为

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^P \left(\frac{a_n - ib_n}{2} e^{in\omega x} + \frac{a_n + ib_n}{2} e^{-in\omega x} \right) \quad (3-1-4)$$

其中, $\cos(\theta) = (e^{i\theta} + e^{-i\theta})/2$, $\sin(\theta) = -i(e^{i\theta} - e^{-i\theta})/2$ 。若记 $c_0 = a_0/2$,当 $n = 1, 2, 3, \dots, P$ 时,记:



第5集
微课视频

$$\begin{cases} c_n = \frac{a_n - ib_n}{2} \\ c_{2P+1-n} = \frac{a_n + ib_n}{2} \end{cases} \quad (3-1-5)$$

其中,一个傅里叶基对应 2 个变量 (a_n 和 b_n), P 个傅里叶基对应 $2P$ 个变量,加上变量 a_0 ,共 $(2P+1)$ 个变量。由于 $e^{i\omega(N-n)x} = e^{-i\omega nx}$,当 $N=2P+1$ 时,式(3-1-4)可以进一步改写为

$$f(x) = \sum_{n=0}^{N-1} c_n \times e^{i\omega nx}, \quad x=0,1,2,\dots,N-1 \quad (3-1-6)$$

将式(3-1-6)展开:

$$\begin{cases} f(x=0) = c_0 \times e^0 + c_1 \times e^{i0\omega \times 1} + c_2 \times e^{i0\omega \times 2} + \dots + c_{N-1} \times e^{i0\omega \times (N-1)} \\ f(x=1) = c_0 \times e^0 + c_1 \times e^{i1\omega \times 1} + c_2 \times e^{i1\omega \times 2} + \dots + c_{N-1} \times e^{i1\omega \times (N-1)} \\ f(x=2) = c_0 \times e^0 + c_1 \times e^{i2\omega \times 1} + c_2 \times e^{i2\omega \times 2} + \dots + c_{N-1} \times e^{i2\omega \times (N-1)} \\ \vdots \\ f(x=N-1) = c_0 \times e^0 + c_1 \times e^{i(N-1)\omega \times 1} + c_2 \times e^{i(N-1)\omega \times 2} + \dots + c_{N-1} \times e^{i(N-1)\omega \times (N-1)} \end{cases} \quad (3-1-7)$$

矩阵表示为

$$\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(N-1) \end{bmatrix} = \mathbf{T} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{bmatrix} = \mathbf{T} \cdot \mathbf{c} \quad (3-1-8)$$

其中,

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{i1\omega \times 1} & e^{i1\omega \times 2} & \dots & e^{i1\omega \times (N-1)} \\ 1 & e^{i2\omega \times 1} & e^{i2\omega \times 2} & \dots & e^{i2\omega \times (N-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & e^{i(N-1)\omega \times 1} & e^{i(N-1)\omega \times 2} & \dots & e^{i(N-1)\omega \times (N-1)} \end{bmatrix} \in \mathbf{C}^{N \times N} \quad (3-1-9)$$

以上是傅里叶级数的介绍。傅里叶变换是将信号从空域变换到频域,获得组成该信号的波的频率和幅值,本质上就是求解式(3-1-8)中的 \mathbf{c} ,即

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{bmatrix} = \mathbf{T}^{-1} \cdot \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(N-1) \end{bmatrix} \quad (3-1-10)$$

其中,

$$\mathbf{T}^{-1} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-i\omega \times 1} & e^{-i\omega \times 2} & \cdots & e^{-i\omega \times (N-1)} \\ 1 & e^{-i2\omega \times 1} & e^{-i2\omega \times 2} & \cdots & e^{-i2\omega \times (N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-i(N-1)\omega \times 1} & e^{-i(N-1)\omega \times 2} & \cdots & e^{-i(N-1)\omega \times (N-1)} \end{bmatrix} \in \mathbf{C}^{N \times N} \quad (3-1-11)$$

则信号长度为 N 的非周期信号 $f(x)$, $\omega = 2\pi/N$, 其傅里叶变换为

$$F(n) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \times e^{-i\omega n x}, \quad n = 0, 1, 2, \dots, N-1 \quad (3-1-12)$$

同理, 傅里叶逆变换为

$$f(x) = \sum_{n=0}^{N-1} F(n) \times e^{i\omega n x}, \quad x = 0, 1, 2, \dots, N-1 \quad (3-1-13)$$

3.1.2 2D 傅里叶变换

根据前面的介绍, 针对 2D 图像数据 $f(x, y)$, 图像大小为 $M \times N$, 其傅里叶变换和傅里叶逆变换可以表示为

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(ux/M + vy/N)} \quad (3-1-14)$$

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi(ux/M + vy/N)}$$

其中, 系数 $1/(MN)$ 也可以放在傅里叶逆变换中, $F(u, v)$ 是 $f(x, y)$ 的傅里叶变换, $F(u, v)$ 的复数表示形式和指数表示形式分别为 $F(u, v) = R(u, v) + iI(u, v)$ 和 $F(u, v) = |F(u, v)| e^{i\Phi(u, v)}$, 相角和频谱的定义如下:

$$\begin{cases} \Phi(u, v) = \arctan\left(\frac{I(u, v)}{R(u, v)}\right) \\ |F(u, v)| = \sqrt{I^2(u, v) + R^2(u, v)} \end{cases} \quad (3-1-15)$$

而功率谱为 $E(u, v) = |F(u, v)|^2$ 。下面给出图像傅里叶变换的 MATLAB 编程实现代码, 基于该代码, 图像傅里叶变换的结果如图 3-1-1 所示。

```

clc;close all;clear all;
Centering = 0; % 中心化处理
img = double(imread('gray.tif'));
[M,N] = size(img);
F_I = zeros(M,N);
%% 傅里叶变换
for u = 0:M-1
    for v = 0:N-1
        s = 0;
        for x = 0:M-1
            for y = 0:N-1
                tmp = -1i * 2 * pi * (u * x/M + v * y/N);
                if Centering == 1
                    s = s + img(x+1,y+1) * exp(tmp) * (-1)^(x+y);
                else

```

```

        s = s + img(x+1,y+1) * exp(tmp);
    end
end
end
s = s/(M*N);
F_I(u+1,v+1) = s;
end
end
figure,imshow(10 * log(abs(F_I)),[]); % 灰度拉伸,显示频谱图
%% 傅里叶逆变换
Img = zeros(M,N);
for x=0:M-1
    for y=0:N-1
        s = 0;
        for u=0:M-1
            for v=0:N-1
                tmp = 1i * 2 * pi * (u * x/M + v * y/N);
                s = s + F_I(u+1,v+1) * exp(tmp);
            end
        end
        if Centering == 1
            Img(x+1,y+1) = s * (-1)^(x+y);
        else
            Img(x+1,y+1) = s;
        end
    end
end
end
figure,imshow(real(Img),[]);

```

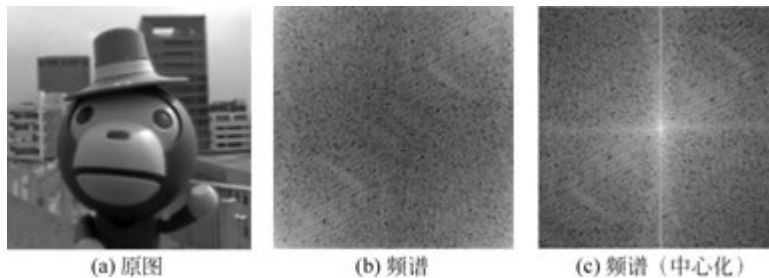


图 3-1-1 图像傅里叶变换的结果

3.1.3 傅里叶变换的意义分析

根据式(3-1-14),结合欧拉公式,图像的傅里叶变换可以表示为

$$\begin{aligned}
 F(u,v) = & \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \times \cos\left(\frac{2\pi ux}{M} + \frac{2\pi vy}{N}\right) - \\
 & i \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \times \sin\left(\frac{2\pi ux}{M} + \frac{2\pi vy}{N}\right)
 \end{aligned} \quad (3-1-16)$$

根据式(3-1-16),可以直观感受到频域中任一点 $F(u,v)$,其实部是由图像 $f(x,y)$ 与余弦模板卷积获得;其虚部是由图像 $f(x,y)$ 与正弦模板卷积获得;卷积核的大小与图

像大小一致。如图 3-1-2 所示为 8×8 大小的图像对应的卷积模板：每个 (u, v) 对应一个 8×8 的卷积核模板，共有 64 个 8×8 的模板，且满足对称关系（如 $(u=2, v=3)$ 对应的模板和 $(u=3, v=2)$ 对应模板是互为转置的）。 $(u=0, v=0)$ 对应的是全 1 的模板， $F(0, 0)$ 是整幅图像像素值的均值。傅里叶变换本质意义就是通过与不同模式（频率）的卷积核做卷积操作，提取图像中该模式的幅值。图 3-1-2(c) 所示为中心化（模板做中心化平移）之后的模板，中心列（第 4 列）对应提取水平方向的图像纹理，若图像中包含大部分水平方向的边缘，则频谱图像中心列的幅值比较大（如图 3-1-1(c)）；中心行（第 4 行）对应提取垂直方向的图像纹理，若图像中包含大部分垂直方向的边缘，则频谱图像中心行的幅值比较大。

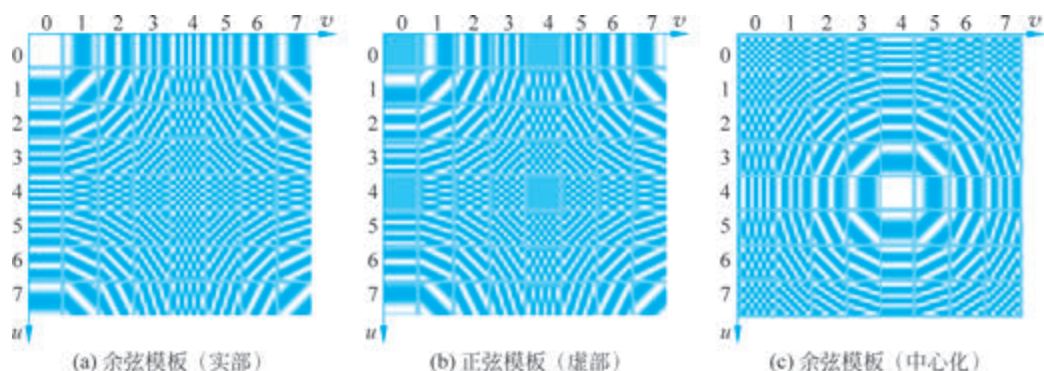


图 3-1-2 图像傅里叶变换的卷积模板

3.1.4 傅里叶变换的性质

傅里叶变换是一种可分离的正交变换，具有可分离性、平移性和旋转性等性质，下面详细介绍比较常用的性质。

(1) 可分离性：2D 傅里叶变换可以分解为“先行后列”或“先列后行”的 2 个 1D 傅里叶变换，逆变换也是如此。

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} e^{-i\frac{2\pi ux}{M}} \cdot \sum_{y=0}^{N-1} f(x, y) e^{-i\frac{2\pi vy}{N}} \quad (3-1-17)$$

(2) 平移性： $f(x, y)$ 在空域平移相当于把其变换到频域与一个线性指数项相乘； $f(x, y)$ 在空域与一个指数项相乘相当于把其变换到频域平移。

$$\begin{aligned} f(x - u_0, y - v_0) &\leftrightarrow F(u, v) e^{-i2\pi\left(\frac{u_0 u}{M} + \frac{v_0 v}{N}\right)} \\ f(x, y) e^{i2\pi\left(\frac{u_0 x}{M} + \frac{v_0 y}{N}\right)} &\leftrightarrow F(u - u_0, v - v_0) \end{aligned} \quad (3-1-18)$$

频谱中心化：频域平移量为 $(u_0 = M/2, v_0 = N/2)$ ，则对应的空域图像 $f(x, y)$ 相乘 $(-1)^{x+y}$ ，频谱中心化的结果如图 3-1-1(c) 所示。频谱中心化之前的低频分量（图像平滑区域）主要分布在频谱的四角，中心化之后，图像的低频分量集中于频谱中心，便于后续的图像增强和去噪处理。因此，频谱中心化常作为图像频域变换的基本操作，在本书后面章节中，未做特殊说明的情况下，傅里叶变换均做了中心化处理。

(3) 旋转性：借助极坐标变换 $x = r \cos \theta, y = r \sin \theta, u = w \cos \varphi, v = w \sin \varphi$ ，将 $f(x, y)$

和 $F(u, v)$ 转换为 $f(r, \theta)$ 和 $F(\omega, \varphi)$ 。对 $f(r, \theta)$ 旋转 θ_0 , 对应将其傅里叶变换也旋转 θ_0 , 反之亦然, 如图 3-1-3 所示。

$$f(r, \theta + \theta_0) \leftrightarrow F(\omega, \varphi + \theta_0) \quad (3-1-19)$$

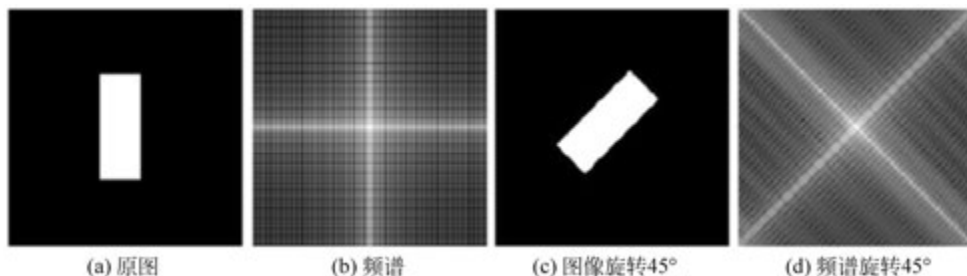


图 3-1-3 图像傅里叶变换旋转性质示例

傅里叶变换的其他性质如下所示。

线性特性:

$$F\{f_1(x, y) + f_2(x, y)\} = F\{f_1(x, y)\} + F\{f_2(x, y)\} \quad (3-1-20)$$

比例特性:

$$\begin{cases} af(x, y) \leftrightarrow aF(u, v) \\ f(ax, by) \leftrightarrow \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right) \end{cases} \quad (3-1-21)$$

卷积特性:

$$F\{f_1(x, y) \otimes f_2(x, y)\} = F\{f_1(x, y)\} \cdot F\{f_2(x, y)\} \quad (3-1-22)$$



第 6 集
微课视频

3.1.5 快速傅里叶变换

傅里叶变换所需的计算量很大, 直接进行一个 $N \times N$ 大小图像的 2D 傅里叶变换需要 N^4 次复数乘法运算和 $N^2(N^2 - 1)$ 次复数加法运算。因此, 快速傅里叶变换被提出, 用于减少计算量。实际中根据傅里叶变换的可分离性, 可以只考虑 1D 快速傅里叶变换。1D 傅里叶变换和傅里叶逆变换的公式可以进一步改写为

$$\begin{cases} F(n) = \sum_{x=0}^{N-1} f(x) \times e^{-i\frac{2\pi nx}{N}} = \sum_{x=0}^{N-1} f(x) \times W_N^{nx} \\ f(x) = \frac{1}{N} \sum_{n=0}^{N-1} F(n) \times e^{i\frac{2\pi nx}{N}} = \frac{1}{N} \sum_{n=0}^{N-1} F(n) \times W_N^{-nx} \end{cases} \quad (3-1-23)$$

其中, $n=0, 1, 2, \dots, N-1, x=0, 1, 2, \dots, N-1, W_N^{xn} = e^{-i2\pi\frac{xn}{N}}$ 。W 满足以下性质:

$$\begin{cases} W_N^{(-n)x} = W_N^{(N-n)x} \\ W_N^{(x+N/2)} = -W_N^x \\ W_N^{N/2} = -1 \end{cases} \quad (3-1-24)$$

将 $f(x)$ 的奇偶项进行分离, 并记 $x=2r$ 和 $x=2r+1$, 则 1D 傅里叶变换可以表示为

$$F(n) = \sum_{r=0}^{N/2-1} f(2r)W_N^{2nr} + \sum_{r=0}^{N/2-1} f(2r+1)W_N^{(2r+1)n}$$

$$\begin{aligned}
 &= \sum_{r=0}^{N/2-1} f_e(r) W_{N/2}^{rn} + W_N^n \cdot \sum_{r=0}^{N/2-1} f_o(r) W_{N/2}^{rn} \\
 &= F_e(n) + W_N^n \cdot F_o(n)
 \end{aligned} \tag{3-1-25}$$

此时 $n \in [0, N/2-1]$ 。当 $n \in [N/2, N-1]$ 时, 傅里叶变换为

$$\begin{cases} F_e\left(n + \frac{N}{2}\right) = \sum_{r=0}^{N/2-1} f_e(r) W_{N/2}^{r(n+N/2)} = \sum_{r=0}^{N/2-1} f_e(r) W_{N/2}^{rn} = F_e(n) \\ F_o\left(n + \frac{N}{2}\right) = F_o(n) \end{cases} \tag{3-1-26}$$

根据 $F(n) = F_e(n) + W_N^n \cdot F_o(n)$ 和式(3-1-26), 可得

$$\begin{aligned}
 F\left(n + \frac{N}{2}\right) &= F_e\left(n + \frac{N}{2}\right) + W_N^{n+N/2} \cdot F_o\left(n + \frac{N}{2}\right) \\
 &= F_e(n) - W_N^n \cdot F_o(n)
 \end{aligned} \tag{3-1-27}$$

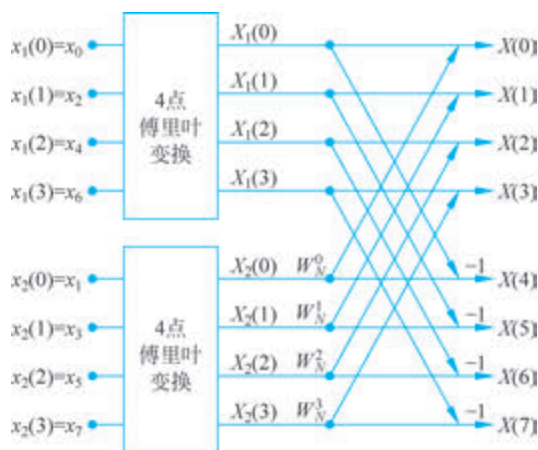
则 1D 快速傅里叶变换的蝶形变换公式为

$$\begin{cases} F(n) = F_e(n) + W_N^n \cdot F_o(n) \\ F\left(n + \frac{N}{2}\right) = F_e(n) - W_N^n \cdot F_o(n) \end{cases}, \quad n \in [0, N/2-1] \tag{3-1-28}$$

快速傅里叶变换核心思想: 将原信号分为奇数项和偶数项, 通过不断的一个奇数一个偶数的相加(减), 最终得到傅里叶变换的结果。

【例 3-1-1】 信号长度为 8 的信号 $\mathbf{x} = [x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T, N=8=2^3$, 可以进行 3 次奇偶分解。

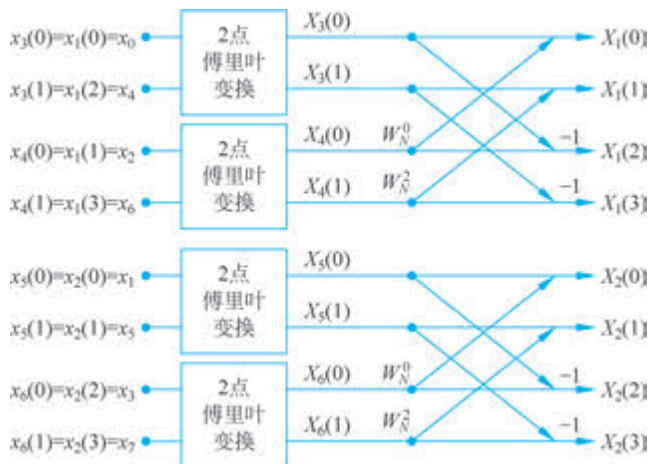
首先根据蝶形变换式(3-1-28), 进行第一次奇偶分解:



其次, 4 点傅里叶变换可以进一步分解为 2 个 2 点傅里叶变换, 以 $x_1(r)$ 为例, 令 $x_1(2l) = x_3(l)$ 和 $x_1(2l+1) = x_4(l)$, 则表达如下:

$$\begin{aligned}
 X_1(n) &= \sum_{r=0}^{N/2-1} x_1(r) W_{N/2}^{nr} = \sum_{l=0}^{N/4-1} x_3(l) W_{N/4}^{ln} + W_N^{n/2} \cdot \sum_{l=0}^{N/4-1} x_4(l) W_{N/4}^{ln} \\
 &= X_3(n) + W_N^{n/2} \cdot X_4(n)
 \end{aligned}$$

同理可以求出 $X_2(n)$ 。

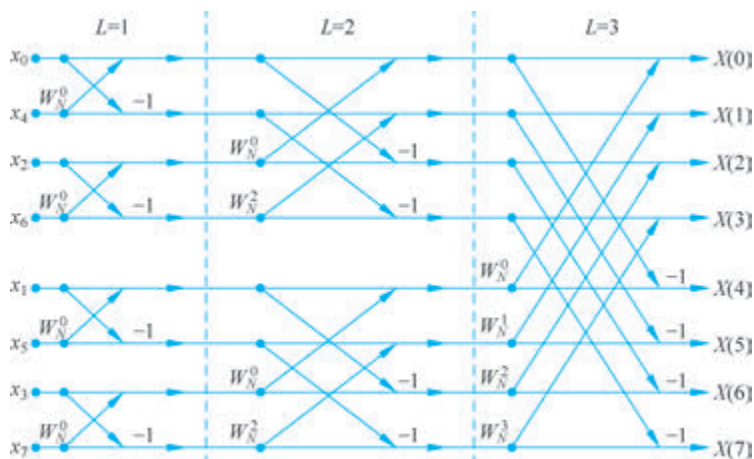


因为 $X_3(n) = \sum_{l=0}^{N/4-1} x_3(l)W_{N/4}^{ln} = \sum_{l=0}^1 x_3(l)W_2^{ln}$, 则可以求得

$$X_3(0) = x_3(0)W_2^0 + x_3(1)W_2^0 = x_0 + W_N^0 \cdot x_4$$

$$X_3(1) = x_3(0)W_2^1 + x_3(1)W_2^1 = x_0 - W_N^0 \cdot x_4$$

同理可以求得 $X_4(n)$, $X_5(n)$ 和 $X_6(n)$ 。该信号整体傅里叶变换可以形象表示如下：



长度为 N 的 1D 信号, 直接进行傅里叶变换的计算量为 N^2 次复数乘法运算和 $N(N-1)$ 次复数加法运算。结合上面的例子, 当 $N=2^L$ 时, 快速傅里叶变换共有 L 级蝶形运算, 每级都由 $N/2$ 个蝶形运算组成, 而每个蝶形运算有 1 次复数乘法和 2 次复数加法, 因此每级运算都需 $N/2$ 次复数乘法和 N 次复数加法。这样 L 级运算共需 $N/(2 \cdot \lg N)$ 次复数乘法运算和 $N \cdot \lg N$ 次复数加法运算。

下面介绍如何编程实现快速傅里叶变换, 主要涉及排序和蝶形运算, 下面对排序策略进行介绍。首先, 可以按照“按位倒序规则”进行信号排序, 如图 3-1-4 所示: 信号长度为 8, 可以采用 3 位进行编码, 然后将编码地址进行倒序, 获得新的顺序, 比如“100”, 倒序为“001”。

| 原地址 | 原顺序 | 新地址 | 新顺序 |
|-----|------|-----|------|
| 000 | x(0) | 000 | x(0) |
| 001 | x(1) | 100 | x(4) |
| 010 | x(2) | 010 | x(2) |
| 011 | x(3) | 110 | x(6) |
| 100 | x(4) | 001 | x(1) |
| 101 | x(5) | 101 | x(5) |
| 110 | x(6) | 011 | x(3) |
| 111 | x(7) | 111 | x(7) |

图 3-1-4 按位倒序

下面给出图像快速傅里叶变换的 MATLAB 编程实现代码。

```

clc;close all;clear all;
img = double(imread('gray.tif'));
[M,N] = size(img);
M1 = 2^(floor(log2(M))); N1 = 2^(floor(log2(N)));
img = imresize(img,[M1,N1],'bicubic');
figure,imshow(img,[]);
img_ori = img;
Centering = 1; % 中心化处理
if Centering==1
    tmp_x = 0:M1-1;
    tmp_y = 0:N1-1;
    [X,Y] = meshgrid(tmp_x,tmp_y);
    X = X';
    Y = Y';
    img = img.*(-1).^(X+Y);
end
%% FFT
F = zeros(M1,N1);
for i = 1:M1
    y = img(i,:);
    f = FFT_ours(y,0);
    F(i,:) = f;
end
img_tmp = F;
for j = 1:N1
    y = img_tmp(:,j);
    f = FFT_ours(y,0);
    F(:,j) = f;
end
figure,imshow(10*log(abs(F)),[]);
%% Inverse FFT
img_out = zeros(M1,N1);
for i = 1:M1
    y = F(i,:);
    f = FFT_ours(y,1);
    img_out(i,:) = f;
end

```

```

img_tmp = img_out;
for j = 1:N1
    y = img_tmp(:,j);
    f = FFT_ours(y,1);
    img_out(:,j) = f;
end
img_out = real(img_out);
if Centering == 1
    img_out = img_out .* (-1).^(X+Y);
end
figure, imshow(abs(img_out),[]);

% 1D 快速傅里叶变换
function F = FFT_ours(y,inv)
L = length(y);
%% 排序
C = log2(L);
Ind = zeros(L,1);
for i = 1:L
    a = dec2bin(i-1,C);
    b = fliplr(a);
    Ind(i) = bin2dec(b)+1;
end
y_1 = y(Ind);
F = y_1;
%%
for layer = 1:C
    N = 2^(layer-1);
    F_last = F;
    for j = 1:N
        for i = j:2*N:L-N
            F(i) = (F_last(i) + W(2*N,mod(i-1,N),inv) * F_last(i+N));
            F(i+N) = (F_last(i) - W(2*N,mod(i-1,N),inv) * F_last(i+N));
        end
    end
end
if inv == 1
    F = (F)/L;
end
end

% W 函数
function out = W(N,u,inv)
if inv == 1
    c = 1;
else
    c = -1;
end
out = exp(c * 1i * 2 * pi * u/N);
end

```

3.2 离散余弦变换

离散余弦变换(Discrete Cosine Transform, DCT)是一种可分离的正交变换,在图像压缩领域得到了广泛应用。2D的离散余弦正逆变换定义如下:

$$\begin{cases} F_c(\mu, v) = \frac{2}{\sqrt{MN}} c(\mu) c(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{\pi}{2N} (2x+1)\mu \right] \cos \left[\frac{\pi}{2M} (2y+1)v \right] \\ f(x, y) = \frac{2}{\sqrt{MN}} \sum_{\mu=0}^{M-1} \sum_{v=0}^{N-1} c(\mu) c(v) F_c(\mu, v) \cos \left[\frac{\pi}{2N} (2x+1)\mu \right] \cos \left[\frac{\pi}{2M} (2y+1)v \right] \end{cases} \quad (3-2-1)$$

其中,

$$c(t) = \begin{cases} \frac{1}{\sqrt{2}}, & t=0 \\ 1, & t=1, 2, \dots, N-1 \end{cases} \quad (3-2-2)$$

3.3 Gabor 变换

傅里叶变换常用于图像频域处理,然而该变换只能获取频率成分,不能提供各频率成分出现的时刻。Gabor 变换是 Dennis Gabor 于 1946 年提出的,其基本思想是把信号分成若干小的时间间隔,再用傅里叶变换分析每一个时间间隔,以确定信号在该时间间隔的频率成分。

二维 Gabor 滤波器能够很好地描述对应于空间频率、空间位置及方向选择性的局部结构信息。Gabor 滤波器的频率和方向表示接近人类视觉系统对于频率和方向的表示,常用于图像纹理的表示和描述。二维 Gabor 滤波器还对图像的边缘敏感,能够提供良好的方向选择和尺度选择特性,并且对光照变化不敏感,能够提供对光照变化良好的适应性。二维 Gabor 滤波器的定义如下:

$$G_{uv}(x, y) = \frac{k_u^2}{\sigma^2} \exp\left(-\frac{k_u^2(x^2 + y^2)}{2\sigma^2}\right) \times \left[\exp(i(k_u \cos\phi_v x + k_u \sin\phi_v y)) - \exp\left(-\frac{\sigma^2}{2}\right) \right] \quad (3-3-1)$$

其中, $k_u = 2^{-(u+2)/2} \pi$, $\phi_v = v\pi/K$ 。 u 的取值决定 Gabor 滤波的波长(尺度信息), v 表示 Gabor 滤波的方向, K 表示滤波器的总方向数量。下面给出二维 Gabor 滤波器的 MATLAB 编程实现代码,实部和虚部滤波器可视化如图 3-3-1 所示。

```
clc;close all;clear all;
GaborH = 31; % Gabor 滤波器高度
GaborW = 31; % Gabor 滤波器宽度
sigma = sqrt(2) * pi;
unum = 5; % 波长数量
vnum = 8; % 方向数
```



第 7 集
微课视频

```

N = vnum * unum; % Gabor kernel's number, default (5 scales and 8 orientations)
%%
GaborReal = zeros(GaborH, GaborW, N); % 实部滤波器
GaborImg = zeros(GaborH, GaborW, N); % 虚部滤波器
for u = 0:unum-1
    for v = 0:vnum-1
        [GaborReal(:, :, u * vnum + v + 1), GaborImg(:, :, u * vnum + v + 1)] = ...
            MakeGaborKernal(GaborH, GaborW, u, v, sigma, unum);
    end
end

function [GaborReal, GaborImg] = MakeGaborKernal(GaborH, GaborW, U, V, sigma, K)
HarfH = fix(GaborH/2);
HarfW = fix(GaborW/2);
Qv = pi * V/K;
sqsigma = sigma * sigma;
Ku = pi * (2^(-(U+2)/2));
postmean = exp(-sqsigma/2);
for i = -HarfH : HarfH
    for j = -HarfW : HarfW
        tmp1 = exp(-(Ku * Ku * (i * i + j * j))/(2 * sqsigma));
        tmp2 = cos(Ku * cos(Qv) * i + Ku * sin(Qv) * j) - postmean;
        tmp3 = sin(Ku * cos(Qv) * i + Ku * sin(Qv) * j);
        GaborReal(i + HarfH + 1, j + HarfW + 1) = Ku * Ku * tmp1 * tmp2/sqsigma;
        GaborImg(i + HarfH + 1, j + HarfW + 1) = Ku * Ku * tmp1 * tmp3/sqsigma;
    end
end
end

```

如图 3-3-1 所示, Gabor 滤波器可以提取图像中不同尺度下不同方向的特征, 以实部模板(如图 3-3-1(a)所示)为例, 对图 3-3-2(a)做卷积操作, 可以获得不同的图像特征, 如图 3-3-2(b)所示。

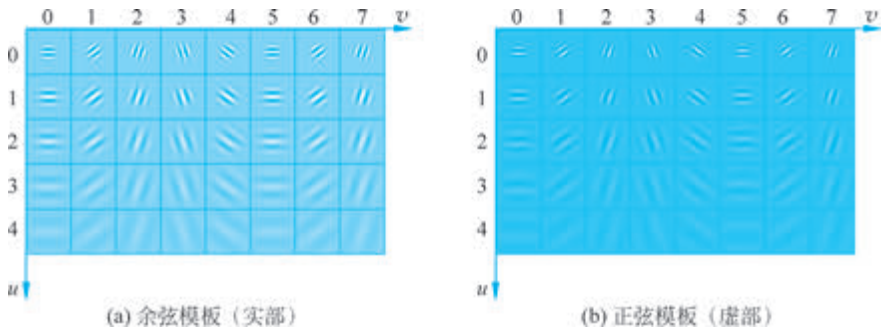


图 3-3-1 Gabor 滤波器(3 个尺度, 5 个方向)

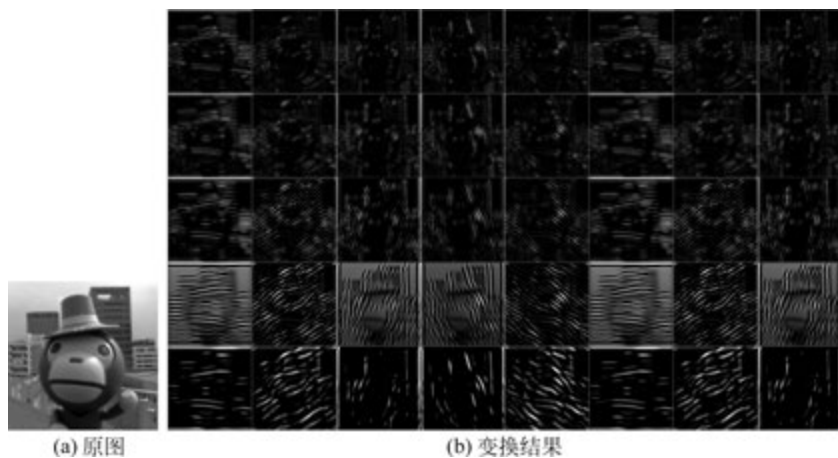


图 3-3-2 Gabor 滤波器的滤波结果

3.4 小波变换

根据前面的分析可知,傅里叶变换只能获取频率成分,不能提供各频率成分出现的时刻。Gabor 变换可以看作加窗的傅里叶变换,同时提供时频信息,然而理论证明,Gabor 变换的窗口宽度和高度的乘积为定值,即窗口大小对所有频率是固定不变的,窗太窄,窗内的信号太短,会导致频率分析不够精准,频率分辨率差。窗太宽,时域上又不够精细,时间分辨率低。

小波变换可以根据频率大小自动调节窗口大小,是一种自适应的时频分析方法,可以进行多分辨率分析。连续信号 $f(t)$ 的小波变换表示为

$$W_f(a, b) = \langle f(t), \psi_{a,b} \rangle = \frac{1}{\sqrt{|a|}} \int_{\mathbf{R}} f(t) \psi^* \left(\frac{t-b}{a} \right) dt \quad (3-4-1)$$

其中, $\langle \cdot \rangle$ 表示内积操作, ψ^* 为 ψ 的复共轭。 $a, b \in \mathbf{R}$ 且 $a \neq 0$ 。 $\psi_{a,b}$ 为小波函数, a 为尺度因子,控制小波函数的伸缩; b 为平移因子,控制小波函数的平移。 $\psi_{a,b}(t)$ 的定义如下:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi \left(\frac{t-b}{a} \right) \quad (3-4-2)$$

将式(3-4-2)中的小波函数离散化:尺度因子取为 2^m ,平移因子为 $2^m \times n$, m 和 n 均为整数,则离散化的小波函数为二进小波,表示为

$$\psi_{m,n}(t) = \frac{1}{\sqrt{2^m}} \psi \left(\frac{t}{2^m} - n \right) \quad (3-4-3)$$

3.4.1 Haar 小波分析

下面介绍图像处理中常见的 Haar 小波函数。在介绍 Haar 小波之前,首先给出 Haar 尺度函数的定义: $\phi_{j,k}(t) = \phi(2^j t - k)$, 其中, $k = 0, 1, 2, \dots, 2^j - 1$, 且 $\phi(t)$ 的定义如下:

$$\phi(t) = \begin{cases} 1, & 0 \leq t < 1 \\ 0, & \text{其他} \end{cases} \quad (3-4-4)$$

Haar 尺度函数对应的波形如图 3-4-1 所示。

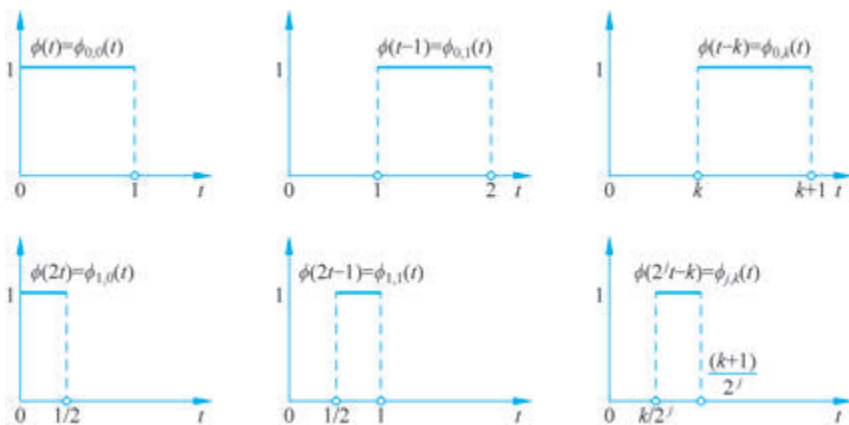


图 3-4-1 Haar 尺度函数对应的波形

Haar 小波函数的定义为 $\psi_{j,k}(t) = \phi(2^j t - k)$, 其中 $\phi(t) = \phi(2t) - \phi(2t-1)$ 。则,

$$\begin{aligned} \psi_{j,k}(t) &= \phi(2^j t - k) = \phi(2^{j+1} t - 2k) - \phi(2^{j+1} t - (2k+1)) \\ &= \phi_{j+1,2k}(t) - \phi_{j+1,2k+1}(t) \end{aligned} \quad (3-4-5)$$

下面通过一个例子直观感受一下小波变换。

【例 3-4-1】 序列 $\{x_1, x_2, x_3, x_4\}$ 的小波变换, 可以看作分段的常数序列, 波形如图 3-4-2 所示。

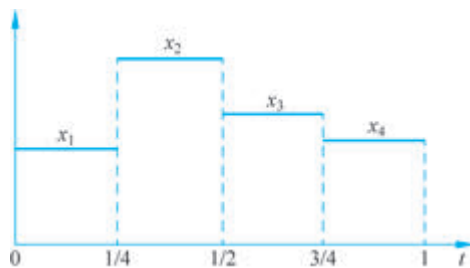


图 3-4-2 序列 $\{x_1, x_2, x_3, x_4\}$ 的小波变换波形

则图 3-4-2 中的波形可以表示为

$$f(t) = x_1 \cdot \phi_{2,0}(t) + x_2 \cdot \phi_{2,1}(t) + x_3 \cdot \phi_{2,2}(t) + x_4 \cdot \phi_{2,3}(t)$$

记 $a_{1,0} = (x_1 + x_2)/2$, $d_{1,0} = (x_1 - x_2)/2$, $a_{1,1} = (x_3 + x_4)/2$, $d_{1,1} = (x_3 - x_4)/2$, 则 $f(t)$ 可以改写为

$$f(t) = a_{1,0} \cdot \phi_{1,0}(t) + a_{1,1} \cdot \phi_{1,1}(t) + d_{1,0} \cdot \psi_{1,0}(t) + d_{1,1} \cdot \psi_{1,1}(t)$$

记 $a_{0,0} = (a_{1,0} + a_{1,1})/2$, $d_{0,0} = (a_{1,0} - a_{1,1})/2$, 则 $f(t)$ 进一步改写为

$$f(t) = a_{0,0} \cdot \phi_{0,0}(t) + d_{0,0} \cdot \psi_{0,0}(t) + d_{1,0} \cdot \psi_{1,0}(t) + d_{1,1} \cdot \psi_{1,1}(t)$$

因此, 序列 $\{x_1, x_2, x_3, x_4\}$ 可由尺度函数和小波函数的系数表示, 即 $\{a_{0,0}, d_{0,0}, d_{1,0},$

$d_{1,1}$ }, 称为小波系数, 表示为

$$\begin{cases} a_{0,0} = \frac{x_1 + x_2 + x_3 + x_4}{4} \\ d_{0,0} = \frac{x_1 + x_2 - x_3 - x_4}{4} \\ d_{1,0} = \frac{x_1 - x_2}{2} \\ d_{1,1} = \frac{x_3 - x_4}{2} \end{cases}$$

结合上面的例子, 长度为 2^n 的序列 $\{x_1, x_2, x_3, \dots\}$, 记为 $\{a_{n,0}, a_{n,1}, \dots, a_{n,2^n-1}\}$, 则对应的波形函数 $f_n(t)$ 可以表示为

$$f_n(t) = a_{n,0} \cdot \phi_{n,0}(t) + a_{n,1} \cdot \phi_{n,1}(t) + \dots + a_{n,2^n-1} \cdot \phi_{n,2^n-1}(t) \quad (3-4-6)$$

将 $f_n(t)$ 做一次小波分解, 得

$$\begin{aligned} f_n(t) = & a_{n-1,0} \cdot \phi_{n-1,0}(t) + a_{n-1,1} \cdot \phi_{n-1,1}(t) + \dots + a_{n-1,2^{n-1}-1} \cdot \phi_{n-1,2^{n-1}-1}(t) + \\ & d_{n-1,0} \cdot \psi_{n-1,0}(t) + d_{n-1,1} \cdot \psi_{n-1,1}(t) + \dots + d_{n-1,2^{n-1}-1} \cdot \psi_{n-1,2^{n-1}-1}(t) \end{aligned} \quad (3-4-7)$$

重复分解多次, 可以获得 $f_n(t)$ 在不同尺度下尺度函数和小波函数的展开式, 其中,

$$\begin{cases} a_{n-1,k} = \frac{a_{n,2k} + a_{n,2k+1}}{2} \\ d_{n-1,k} = \frac{a_{n,2k} - a_{n,2k+1}}{2} \end{cases}, \quad k = 0, 1, \dots, (2^{n-1} - 2) \quad (3-4-8)$$

3.4.2 Haar 小波变换的实现算法

将 Haar 尺度函数 $\phi_{j,k}(t) = \phi(2^j t - k)$ 和 Haar 小波函数 $\psi_{j,k}(t) = \psi(2^j t - k)$, 与式(3-4-3) 进行比较, 并将其标准化为

$$\begin{cases} \phi_{j,k}(t) = 2^{j/2} \cdot \phi(2^j t - k) \\ \psi_{j,k}(t) = 2^{j/2} \cdot \psi(2^j t - k) \end{cases} \quad (3-4-9)$$

标准化的物理意义是: 尺度函数和小波函数在不同尺度下具有相同的能量(即波形图围成的面积相等)。 ϕ 和 ψ 标准化后, 第一步, 定义二尺度方程:

$$\begin{cases} \phi(t) = \phi_{0,0}(t) = \frac{1}{\sqrt{2}} \phi(2t) + \frac{1}{\sqrt{2}} \phi(2t-1) \\ \psi(t) = \psi_{0,0}(t) = \frac{1}{\sqrt{2}} \phi(2t) - \frac{1}{\sqrt{2}} \phi(2t-1) \end{cases} \quad (3-4-10)$$

将式(3-4-10)改写成

$$\begin{cases} \phi(t) = \sum_{k=0}^1 \frac{1}{\sqrt{2}} \phi(2t-k) \\ \psi(t) = \sum_{k=0}^1 (-1)^k \frac{1}{\sqrt{2}} \phi(2t-k) \end{cases} \quad (3-4-11)$$

记 $\mathbf{h} = [h_0, h_1] = [1/\sqrt{2}, 1/\sqrt{2}]$, $\mathbf{g} = [g_0, g_1] = [1/\sqrt{2}, -1/\sqrt{2}]$; 并令 $\bar{\mathbf{h}} = [0, h_1, h_0, 0]$, $\bar{\mathbf{g}} = [0, g_1, g_0, 0]$ 。

第二步,分析滤波器实现信号分解。长度为 2^n 的信号 $a^n = \{a_{n,0}, a_{n,1}, \dots, a_{n,2^n-1}\}$, 分别利用求平均和求差的方法, 可得其低频和低频分量:

$$\begin{cases} a^{n-1} = \{a_{n-1,0}, a_{n-1,1}, \dots, a_{n-1,2^{n-1}}\} \\ d^{n-1} = \{d_{n-1,0}, d_{n-1,1}, \dots, d_{n-1,2^{n-1}}\} \end{cases} \quad (3-4-12)$$

其中,

$$\begin{cases} a_{n-1,k} = \frac{a_{n,2k} + a_{n,2k+1}}{\sqrt{2}} \\ d_{n-1,k} = \frac{a_{n,2k} - a_{n,2k+1}}{\sqrt{2}} \end{cases}, \quad k = 0, 1, \dots, (2^{n-1} - 2) \quad (3-4-13)$$

将原信号补零扩充为无穷序列, 做一维卷积 ($\bar{\mathbf{h}}$ 和 $\bar{\mathbf{g}}$) 操作:

$$\begin{cases} a^n \otimes \bar{\mathbf{h}} = \left\{ \dots, 0, \frac{a_{n,0}}{\sqrt{2}}, \frac{a_{n,0} + a_{n,1}}{\sqrt{2}}, \dots, \frac{a_{n,2^n-2} + a_{n,2^n-1}}{\sqrt{2}}, \frac{a_{n,2^n-1}}{\sqrt{2}}, 0, \dots \right\} \\ a^n \otimes \bar{\mathbf{g}} = \left\{ \dots, 0, -\frac{a_{n,0}}{\sqrt{2}}, \frac{a_{n,0} - a_{n,1}}{\sqrt{2}}, \dots, \frac{a_{n,2^n-2} - a_{n,2^n-1}}{\sqrt{2}}, \frac{a_{n,2^n-1}}{\sqrt{2}}, 0, \dots \right\} \end{cases} \quad (3-4-14)$$

然后, 进行下采样(隔点抽取)操作:

$$\begin{cases} a^{n-1} = D(a^n \otimes \bar{\mathbf{h}}) = \left\{ \dots, 0, \frac{a_{n,0} + a_{n,1}}{\sqrt{2}}, \dots, \frac{a_{n,2^n-2} + a_{n,2^n-1}}{\sqrt{2}}, 0, \dots \right\} \\ d^{n-1} = D(a^n \otimes \bar{\mathbf{g}}) = \left\{ \dots, 0, \frac{a_{n,0} - a_{n,1}}{\sqrt{2}}, \dots, \frac{a_{n,2^n-2} - a_{n,2^n-1}}{\sqrt{2}}, 0, \dots \right\} \end{cases} \quad (3-4-15)$$

第三步,综合滤波器实现信号重构。先对 a^{n-1} 和 d^{n-1} 做(补零)上采样, 再做卷积 (\mathbf{h} 和 \mathbf{g}):

$$\begin{cases} U(a^{n-1}) = \{\dots, 0, a_{n-1,0}, 0, a_{n-1,1}, \dots, 0, a_{n-1,2^{n-1}}, 0, \dots\} \\ U(d^{n-1}) = \{\dots, 0, d_{n-1,0}, 0, d_{n-1,1}, \dots, 0, d_{n-1,2^{n-1}}, 0, \dots\} \end{cases} \quad (3-4-16)$$

$$\begin{cases} U(a^{n-1}) \otimes \mathbf{h} = \left\{ \dots, 0, \frac{a_{n-1,0}}{\sqrt{2}}, \frac{a_{n-1,0}}{\sqrt{2}}, \frac{a_{n-1,1}}{\sqrt{2}}, \dots, \frac{a_{n-1,2^{n-1}}}{\sqrt{2}}, \frac{a_{n-1,2^{n-1}}}{\sqrt{2}}, 0, \dots \right\} \\ U(d^{n-1}) \otimes \mathbf{g} = \left\{ \dots, 0, \frac{d_{n-1,0}}{\sqrt{2}}, -\frac{d_{n-1,0}}{\sqrt{2}}, \frac{d_{n-1,1}}{\sqrt{2}}, \dots, \frac{d_{n-1,2^{n-1}}}{\sqrt{2}}, -\frac{d_{n-1,2^{n-1}}}{\sqrt{2}}, 0, \dots \right\} \end{cases} \quad (3-4-17)$$

则

$$a^n = U(a^{n-1}) \otimes h + U(d^{n-1}) \otimes g \quad (3-4-18)$$

图 3-4-3 所示为小波分解与重构(Mallat 算法)的计算流程。

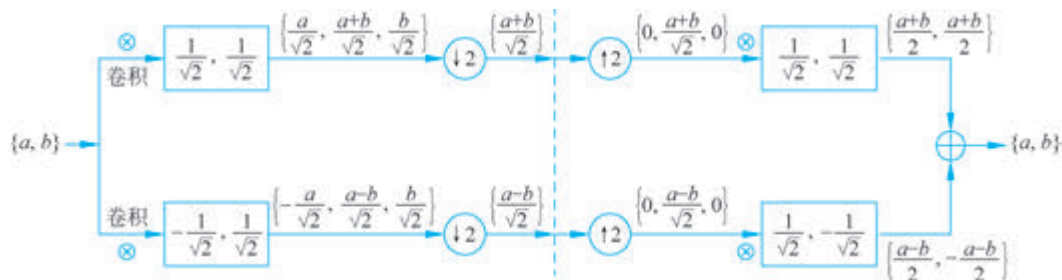


图 3-4-3 小波分解与重构(Mallat 算法)的计算流程

将信号 a^n 一次分解后获得的 a^{n-1} 可以进一步分解,得到 $\{a^{n-2}, d^{n-2}\}$,重复上述步骤,获得不同尺度下的小波分解系数,如图 3-4-4 所示。

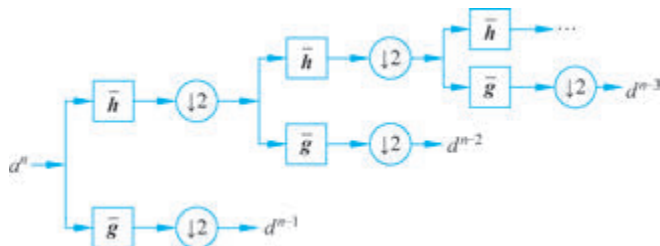


图 3-4-4 小波多尺度分解

3.4.3 图像的小波变换

2D 图像的小波变换可以看作先进行列方向的 1D 小波变换,再对变换的结果进行行方向的 1D 小波变换。首先对符号作如下约定: h_ψ 和 h_φ 分别表示高频和低频滤波器(对应前面的 g 和 h), $h_\psi(-)$ 和 $h_\varphi(-)$ 分别对应前面的 \bar{g} 和 \bar{h} ; 第 $j+1$ 级图像表示为 $W_\varphi(j+1, m, n)$, $W_\psi^H(j+1, m, n)$, $W_\psi^V(j+1, m, n)$ 和 $W_\psi^D(j+1, m, n)$ 。2D 图像的小波分解与重构如图 3-4-5 所示: 低频图像 $W_\varphi(j+1, m, n)$ 经过小波分解,得到 4 个分量,即低频分量 $W_\varphi(j, m, n)$, 包含图像的基本信息; 水平方向的高频分量 $W_\psi^H(j, m, n)$, 反映图像中水平边缘的情况; 垂直方向的高频分量 $W_\psi^V(j, m, n)$, 反映图像中垂直边缘的情况; 高频分量 $W_\psi^D(j, m, n)$ 体现对角方向的高频信息。以上是图像小波变换的简要介绍,更详细的介绍可以参见相关文献^[1]。

如图 3-4-6 所示,图像经过一次小波分解得到的结果与上述分析一致,原图被分解成 1 个低频分量和 3 个高频分量,每部分图像在纵横方向上分别缩小为原来的一半。

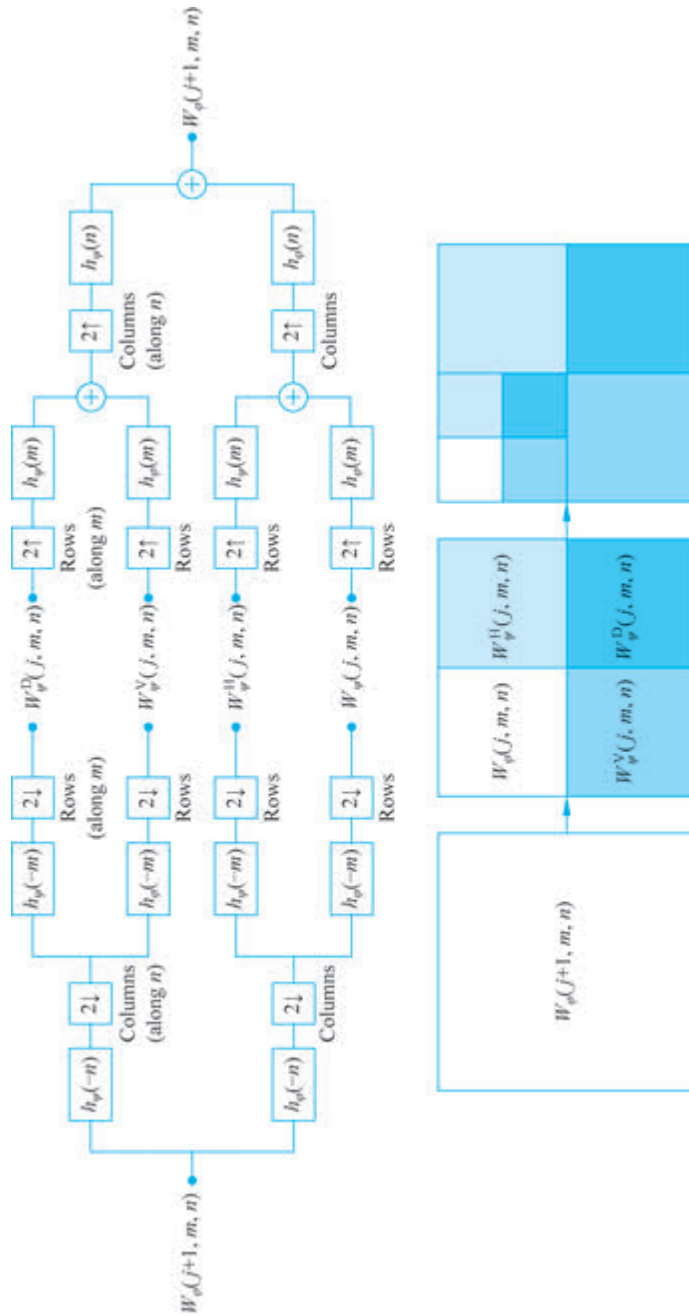


图 3-4-5 2D 图像的小波分解与重构示意

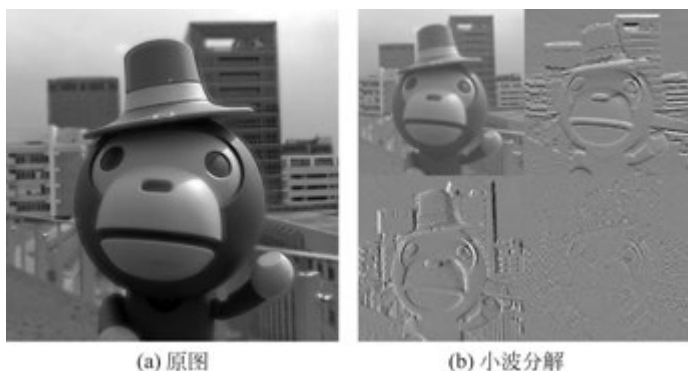


图 3-4-6 图像小波分解结果

3.5 图像变换前沿算法

除了上述几种典型的图像变换,在实际应用中,为了解决图像融合与人脸识别等复杂任务,图像处理领域涌现出了一系列前沿算法。本节将介绍四种重要的算法:主成分分析(PCA)法、矩阵低秩分解法、字典学习法、自动编码器法。

3.5.1 基于主成分分析法的图像变换

主成分分析(PCA)法是一种经典的线性降维技术,旨在通过线性变换将高维数据映射到低维空间,将数据重投影到主成分上,在保留数据主要信息的同时,达到降维和去相关性的目的。PCA 在人脸识别中的发展始于“特征脸(Eigenfaces)^[2]”方法,逐步演化出增强鲁棒性和捕捉非线性特征的改进算法,如 Kernel PCA^[3],并在现代深度学习框架中继续扮演数据预处理与特征分析的重要角色。

主成分分析法的目的是找到一个转换矩阵实现特征降维,可表示为

$$\mathbf{Z} = \mathbf{X}\mathbf{W} \quad (3-5-1)$$

其中 $\mathbf{X} \in \mathbf{R}^{n \times p}$ 是原始数据矩阵,包含 n 个样本,每个样本有 p 个特征, $\mathbf{W} \in \mathbf{R}^{p \times k}$ 是降维矩阵,包含 k 个主成分的特征向量, $\mathbf{Z} \in \mathbf{R}^{n \times k}$ 是降维后的新数据矩阵。为了消除均值对特征向量的影响,对原始矩阵进行数据中心化处理,表示如下:

$$\mathbf{X}' = \mathbf{X} - \boldsymbol{\mu} \quad (3-5-2)$$

协方差矩阵的特征向量表示数据分布的方向,而对应的特征值表示该方向上的变化幅度(方差),计算公式如下:

$$\mathbf{C} = \frac{1}{n-1} \mathbf{X}'^T \mathbf{X}' \quad (3-5-3)$$

$$\mathbf{C} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T \quad (3-5-4)$$

其中, $\boldsymbol{\Lambda}$ 是协方差矩阵的特征值对角矩阵,按降序排列 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M$, \mathbf{V} 是特征向量组成的正交矩阵, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_M]$,其中每个 \mathbf{v}_i 对应一个特征值 λ_i 。若顺序截取前面 k 个最大的特征值 $\lambda_i (i=1, 2, \dots, k) (k < M)$ 对应的特征向量 $\mathbf{v}_i (i=1, 2, \dots, k)$,舍去后面较小的 $(M-k)$ 个特征向量,构成变换矩阵 $\mathbf{W} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$,即达到了将 M 维降为 k 维的目

的(这种方法也称为子空间分解方法^[4]),此处 k 的选择可以依据累积方差贡献率:

$$\text{累积方差贡献率} = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^M \lambda_i} \quad (3-5-5)$$

选择满足累积方差贡献率超过一定阈值的 k 。

3.5.2 基于矩阵低秩分解法的图像变换

基于矩阵低秩分解的图像变换是指将图像数据的矩阵形式分解为表示主要特征的低秩矩阵与表示局部变换的稀疏成分或噪声成分:

$$\mathbf{M} = \mathbf{L} + \mathbf{S} \quad (3-5-6)$$

其中, \mathbf{M} 表示原始图像矩阵, \mathbf{L} 是低秩矩阵, \mathbf{S} 是稀疏矩阵。

为了找到矩阵最紧凑表示,逼近最小秩,需要为该问题的解设置约束条件,低秩矩阵的秩为 $\text{rank}(\mathbf{L})$,稀疏矩阵的稀疏性通过 l_0 表示,约束条件可转换为求解最优解问题,然而直接求解 l_0 与 $\text{rank}(\mathbf{L})$ 属于 NP-hard 问题,为了简化求解,将问题转化为如下形式:

$$\begin{aligned} \min_{\mathbf{L}, \mathbf{S}} \quad & \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \\ \text{s. t.} \quad & \mathbf{M} = \mathbf{L} + \mathbf{S} \end{aligned} \quad (3-5-7)$$

其中, $\|\mathbf{L}\|_*$ 是核范数(Nuclear Norm),核范数是秩的凸松弛近似, $\|\mathbf{S}\|_1$ 是稀疏矩阵 \mathbf{S} 的 L1 范数, λ 是权重。求解最小值可采用交替方向乘子法(ADMM)^[5],迭代求解 \mathbf{L} 和 \mathbf{S} ; 或者奇异值阈值分解^[6],通过软阈值处理奇异值求解 \mathbf{L} 。

3.5.3 基于字典学习法的图像变换

基于字典学习法的图像变换是一种基于稀疏表示理论的先进技术,用于高效处理图像。其核心理念是通过学习一个自适应字典(Dictionary),以稀疏表示的形式重构图像特征,挖掘图像的本质结构和细节信息。

字典学习法的目标是找到一个字典 $\mathbf{D} \in \mathbf{R}^{m \times K}$ 和稀疏表示矩阵 $\mathbf{A} \in \mathbf{R}^{K \times N}$,使得 $\mathbf{X} \approx \mathbf{DA}$ 且每个稀疏系数向量 α_i 是稀疏的(大部分分量为零)。为了实现上述目标,构造目标函数:

$$\min_{\mathbf{D}, \mathbf{A}} \|\mathbf{X} - \mathbf{DA}\|_F^2 + \lambda \|\mathbf{A}\|_1 \quad (3-5-8)$$

其中, $\|\mathbf{X} - \mathbf{DA}\|_F^2 = \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2$ 是重构误差, $\|\mathbf{A}\|_1 = \sum_{i=1}^N \|\alpha_i\|_1$ 是稀疏正则化, λ 是正则化参数。求解上述优化问题,可分为稀疏编码和字典更新两步。

第一步,稀疏编码指固定字典 \mathbf{D} ,优化编码系数 \mathbf{A} ,公式表达如下:

$$\min_{\alpha_i} \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \quad (3-5-9)$$

其中, \mathbf{x}_i 是 m 维图像块, λ 是稀疏性正则化参数,由于 L1 范数具有非光滑性,可采用软阈值法(Soft Thresholding)^[7]来求解,将 \mathbf{x}_i 向字典 \mathbf{D} 上投影,表示为 \mathbf{z}_i ,即 $\mathbf{z} = \mathbf{D}^T \mathbf{x}$,优化函数可表示为

$$\min_{\alpha_i} \frac{1}{2} (\mathbf{z}_i - \alpha_i)^2 + \lambda |\alpha_i| \quad (3-5-10)$$

其中,稀疏正则化项 $\lambda |\alpha_i|$ 在 $\alpha_i = 0$ 处不可微,为求解优化函数最值需要对 α_i 进行分段讨论。

当 $\alpha_i > 0$ 时,令目标导数值为零, $\alpha_i^* = z_i - \lambda$,要求 $\alpha_i > 0$,因此只有当 $z_i > \lambda$ 时,解有效。

当 $\alpha_i < 0$ 时,令目标导数值为零, $\alpha_i^* = z_i + \lambda$,要求 $\alpha_i < 0$,因此只有当 $z_i < -\lambda$ 时,解有效。

当 $\alpha_i = 0$ 时, $\alpha_i^* = 0, |z_i| \leq \lambda$ 。

综上所述可得软阈值函数:

$$\alpha_i^* = \text{sgn}(z_i) \cdot \max(|z_i| - \lambda, 0) \quad (3-5-11)$$

其中, $\text{sgn}(\cdot)$ 是符号函数。

第二步,字典更新是指固定编码系数,优化字典,可表示为优化以下问题:

$$\begin{aligned} \min_{\mathbf{D}} \quad & \| \mathbf{X} - \mathbf{D}\mathbf{A} \|_F^2 \\ \text{s. t.} \quad & \| \mathbf{d}_k \|_2 = 1, \forall k \end{aligned} \quad (3-5-12)$$

其中, \mathbf{d}_k 是字典的第 k 列。字典更新通常采用 K-SVD^[8] 算法,逐列更新 \mathbf{d}_k ,同时调整对应的稀疏系数。

基于矩阵低秩分解法和字典学习法的图像变换结果示意如图 3-5-1 所示,基于矩阵低秩分解理论,可以将场景分解为低秩的背景和稀疏的前景(目标);基于稀疏编码理论,图像块可以用字典与编码系数进行联合表征。

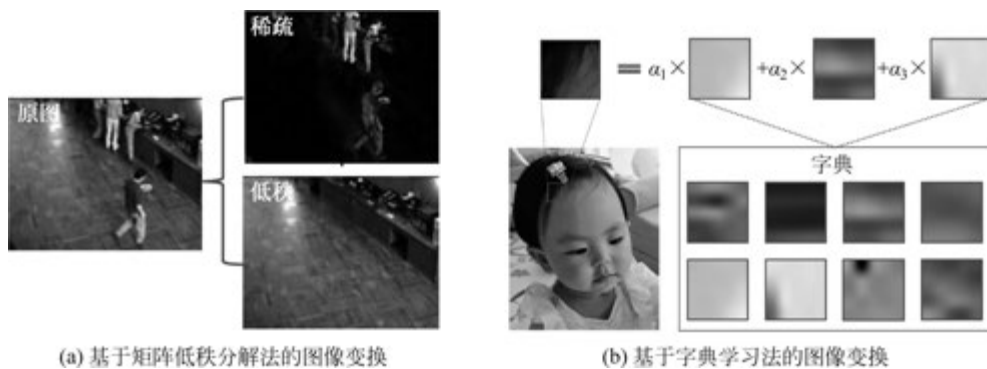


图 3-5-1 基于矩阵低秩分解法和字典学习法的图像变换结果示意

3.5.4 基于自动编码器法的图像变换

自动编码器是一种无监督学习模型,利用神经网络结构实现对图像数据的特征提取和重构,网络结构由两个主要部分组成,分别是编码器(Encoder)与解码器(Decoder)。编码器是将输入数据映射到低维的潜在表示(Latent Space),可表示为

$$\mathbf{z} = f_{\text{encoder}}(\mathbf{x}) = \phi(\mathbf{W}_e \mathbf{x} + b_e) \quad (3-5-13)$$

其中, $\mathbf{x} \in \mathbf{R}^n$ 为输入图像数据,通常展平为一维向量, $\mathbf{z} \in \mathbf{R}^m$ 是潜在表示(编码后的特征),且 $m < n$ 表示特征降维, \mathbf{W}_e, b_e 是编码器的权重和偏置, $\phi(\cdot)$ 是非线性激活函数(如 ReLU、sigmoid)。解码器是从潜在表示重建输入数据,可以去除噪声,补全缺少信息,可表示为

$$\hat{\mathbf{x}} = f_{\text{decoder}}(\mathbf{z}) = \psi(\mathbf{W}_d \mathbf{z} + b_d) \quad (3-5-14)$$

其中, $\hat{x} \in \mathbf{R}^n$ 是重建的输出图像。自动编码器训练的核心目标是使重构后的向量 \hat{x} 尽量还原输入向量 x , 即最小化重构带来的损失, 常用均方误差 (Mean Square Error, MSE) 损失函数表示为

$$\text{Loss} = \|\hat{x} - x\|_2^2 \quad (3-5-15)$$

基于自动编码器变形的有去噪自动编码器 (Denoising Autoencoder)、变分自动编码器 (Variational Autoencoder)、稀疏自动编码器 (Sparse Autoencoder) 和掩码自动编码器 (Masked Autoencoder, MAE) 等, 接下来重点介绍 MAE^[9]。

如图 3-5-2 所示^[9], MAE 编码器将观察到的信号映射到潜在表示, MAE 解码器从潜在表示重建原始信号。与经典的自动编码器不同, MAE 采用了非对称设计, 允许编码器仅对部分观察到的信号进行操作 (无掩码标记), 并采用了轻量级解码器, 该解码器根据潜在表示和掩码标记重新构建完整信号。具体过程如下。① Masking: 将图像划分为规则的非重叠图像块 (Patches), 按照均匀分布对图像块进行随机采样, 并屏蔽 (掩码) 剩余的图像块。② Encoder: 基于一系列 Transformer 块对非掩码的图像块进行编码。③ Decoder: 输入是非掩码图像块的编码和掩码的图像块, 掩码的图像块是共享的和需学习的量, 表明图像该位置需要进行修复。④ Reconstruction target: 通过预测每个掩码图像块的像素值来重建输入, 并只在掩码图像块上计算重建图像与输入图像的 MSE 损失。MAE 具有泛化能力强、训练效率高等优势, MAE 重建的效果如图 3-5-3 所示。

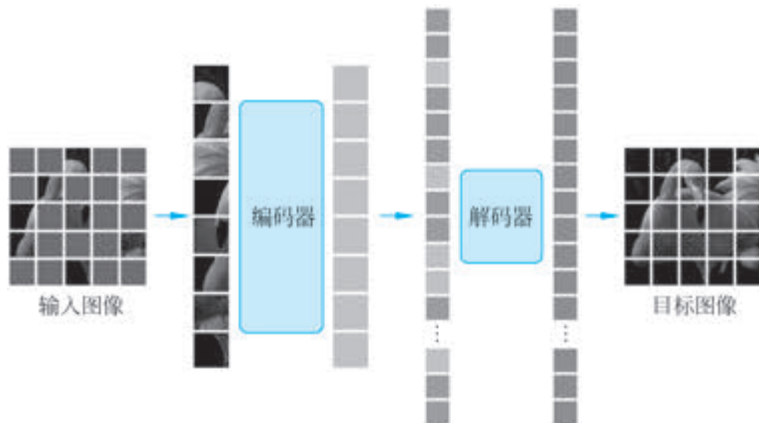


图 3-5-2 MAE 架构

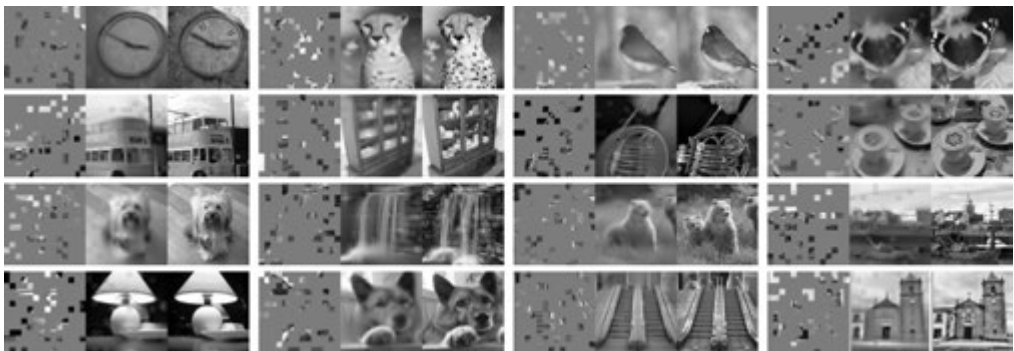


图 3-5-3 MAE 重建效果 (从左至右分别为掩码图像、MAE 重建结果、真实图像)

参考文献

- [1] 孙延奎. 小波变换与图像、图形处理技术[M]. 北京: 清华大学出版社, 2012.
- [2] TURK M A, PENTLAND A P. Face recognition using eigenfaces[C]//Proceedings of 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 1991: 586-591.
- [3] SCHÖLKOPF B, SMOLA A, MÜLLER K R. Nonlinear component analysis as a kernel eigenvalue problem[J]. Neural Computation, 1998, 10(5): 1299-1319.
- [4] OJA E. Subspace methods of pattern recognition[M]. England: Research Studies Press, 1983.
- [5] BOYD S, PARIKH N, CHU E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations and Trends in Machine Learning, 2011, 3(1): 1-122.
- [6] MA S, GOLDFARB D, CHEN L. Fixed point and Bregman iterative methods for matrix rank minimization[J]. Mathematical Programming, 2011, 128(1): 321-353.
- [7] DONOHO D L. De-noising by soft-thresholding[J]. IEEE Transactions on Information Theory, 1995, 41(3): 613-627.
- [8] AHARON M, ELAD M, BRUCKSTEIN A. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation[J]. IEEE Transactions on Signal Processing, 2006, 54(11): 4311-4322.
- [9] HE K, CHEN X, XIE S, et al. Masked autoencoders are scalable vision learners[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2022, 15979-15988.