数字图像处理

图像的变换域及其应用

叶志鹏

南京理工大学泰州科技学院

5th Aug, 2022

1 变换域基础
   - 空间域是什么
   - 变换域是什么
   - 为什么需要变换域
   - Eigen-faces

2 离散傅里叶变换（Discrete Fourier Transform）

3 离散余弦变换（Discrete Cosine Transform）

4 参考文献

空间域是什么

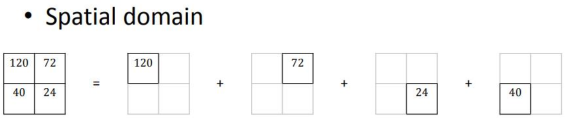


图 1: 2*2 灰度图像的空间域

- 数字图像 $\vec{M}$ 的空间域可以看成标准正交基向量 $\vec{e}_i$ 的线性叠加。

$$\vec{M} = \sum_{i=1}^{N} \lambda_i \vec{e}_i$$

标准正交基

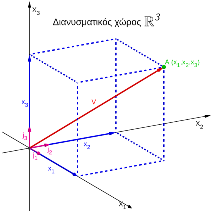


图 2: R3 linear space

- 一组 n 维的线性无关的向量组构成 R^n 的线性空间。只存在全零 a_i 使得 $\sum_{i=0}^{n} a_i \vec{v_i} = 0$。即为线性无关向量组，又称 R^n 空间的基向量。
- 标准正交基的模长为 1，并且两两基向量正交。

$$\forall i,j \quad |\vec{e_i}| = 1 \quad and \quad \vec{e_i} \cdot \vec{e_j} = 0$$

- 如刚才的标准正交基

$$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix},$$

$$\begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}$$

1. 变换域基础
   - 空间域是什么
   - 变换域是什么
   - 为什么需要变换域
   - Eigen-faces
2. 离散傅里叶变换（Discrete Fourier Transform）
3. 离散余弦变换（Discrete Cosine Transform）
4. 参考文献

变换域是什么

- 如果我们换一组正交基会怎么样？

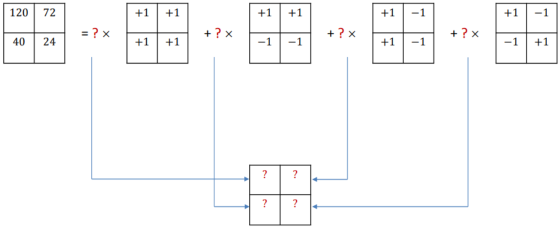


图 3: 一组正交基

变换域是什么

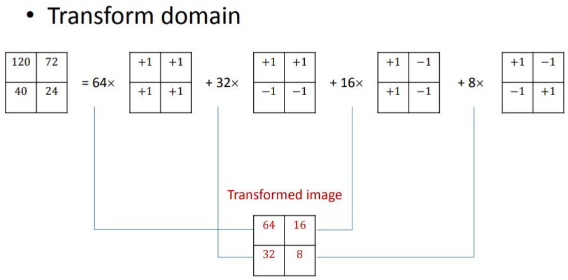


图 4: 变换域的例子

- 变换域就是让空间域的图像在另一组正交基组成的线性空间中换一种表达形式。

变换域是什么

- 我们把空间域 $\vec{x}$ 到变换域 $\vec{p}$ 的映射称为正变换，记为 T

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 120 \\ 72 \\ 40 \\ 24 \end{bmatrix} = \begin{bmatrix} 64 \\ 16 \\ 32 \\ 8 \end{bmatrix}$$

- 变换域 $\vec{p}$ 到空间域 $\vec{x}$ 的映射称为逆变换，记为 T^{-1}

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 64 \\ 16 \\ 32 \\ 8 \end{bmatrix} = \begin{bmatrix} 120 \\ 72 \\ 40 \\ 24 \end{bmatrix}$$

- $T * T^{-1} = E$ 单位阵

1. 变换域基础
   - 空间域是什么
   - 变换域是什么
   - 为什么需要变换域
   - Eigen-faces
2. 离散傅里叶变换（Discrete Fourier Transform）
3. 离散余弦变换（Discrete Cosine Transform）
4. 参考文献

为什么需要变换域

- 变换域使我们能够更灵活地操作数据（例如，过滤、压缩等）
- 提供了让我们理解数据的工具（例如图像分类）
- 不同的应用需要不同的视角去解决问题。

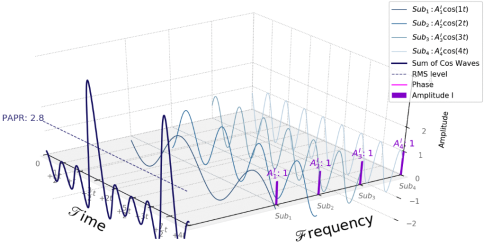


图 5: 时域与频域

Eigen-faces

- 任意一张人脸可以被视为特征人脸的叠加
- 特征人脸构成了人脸空间？

图 6: Eigen faces

如何构建特征脸 Eigen-faces

- 采集人脸数据
- 减去平均脸
- 计算协方差并进行特征值分解得到特征向量（特征脸）

Algorithm 1 PCA Algorithm

1: X: input $n \times d$ data matrix (each row a d-dimensional sample)
2: Normalize the data: subtract mean of X from each row of X
3: Compute the covariance matrix of X to obtain Σ
4: Find eigenvectors and eigenvalues of Σ
5: Compute r eigenvectors with largest eigenvalues to construct the matrix $C_{d\times r}$, where the value of eigenvalues gives importance of each component
6: Transform X using C

$$Y = XC$$

where the number of new dimensional is r ($r \ll d$)

图 7: Principal Component Analysis (PCA) algorithm

特征人脸识别

- 获取人脸；
- 计算出该人脸，对应的特征脸的系数表达（空间域到变换域)；
- 到数据库中查找相近的变换域对应的人的唯一标识（ID)。

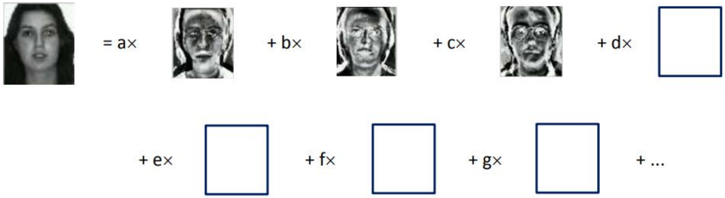


图 8: 计算特征脸的系数

特征人脸识别

Q: 为什么不直接拿人脸图片到数据库找最相似的数据对应的人的信息？
PCA 原理：吴恩达
https://www.bilibili.com/video/BV1By4y1J7A5?p=83
李航：统计学习方法

一维离散傅里叶变换的定义

- 一维离散傅里叶变换有如下定义：对于 N 点序列 $x[n]_{0\leq n\leq N}$，它的离散傅里叶变换（DFT）为

$$x[k]=\sum_{n=0}^{N-1}e^{-i\frac{2\pi}{N}nk}x[n]\quad k=0,1,...,N-1$$

其中 e 是自然对数的底数，i 是虚数单位。通常以符号 $\mathcal{F}$ 表示正变换，即

$$\hat{x}=\mathcal{F}x$$

逆变换（Inverse DFT）为

$$x\left[n\right]=\frac{1}{N}\sum_{k=0}^{N-1}e^{i\frac{2\pi}{N}nk}\hat{x}[k]\qquad n=0,1,\ldots,N-1.$$

可以记为

$$x=\mathcal{F}^{-1}\hat{x}$$

DFT 的数学意义

- 我们先看逆变换 IDFT

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} e^{i\frac{2\pi}{N}nk}\hat{x}[k] \qquad n = 0, 1, \ldots, N-1.$$

可以看成空间域到频域的变换，频域的基向量为
$$u_k = \left[e^{\frac{i2\pi}{N}kn} \,\middle|\, n = 0, 1, \ldots, N-1 \right]^{\mathsf{T}}$$

复指数向量的正交性

$$u_k^T u_{k'}^* = \sum_{n=0}^{N-1} (e^{\frac{2\pi i}{N} kn})(e^{\frac{2\pi i}{N} -k'n}) = \sum_{n=0}^{N-1} e^{\frac{2\pi i}{N}(k-k')n}$$

- 当 $k = k'$ 时，$u_k^T u_{k'}^* = 1 * N$
- 当 $k \neq k'$ 时，$u_k^T u_{k'}^* = 0$ 因为，根据欧拉公式 $e^{ix} = cosx + isinx$。$\int_0^{2\pi} cosx = 0 \quad and \quad \int_0^{2\pi} sinx = 0$

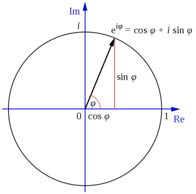


图 9: 欧拉公式-上帝公式

编程验证下正交性

```python
import numpy as np
N = 16
bases = []
for k in range(N):
    basis = []
    for n in range(N):
        value = np.e**(1j*2*np.pi*n*k/N)
        basis.append(value)
    bases.append(basis)
bases = np.array(bases)
vec1 = bases[3]
vec2 = bases[2]
inner_pro = vec1.dot(vec2.conjugate())
print(inner_pro)
```

编程验证下正交性采样 $N = 8$ 时，基向量的图像

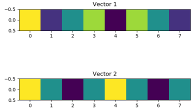


图 10: 两个基向量图像

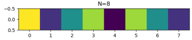


图 12: 两向量点乘过后的图像

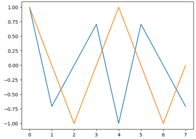


图 11: 两个基向量的曲线

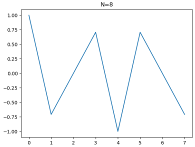


图 13: 两向量点乘过后的曲线

编程验证下正交性采样 $N = 32$ 时，基向量的图像

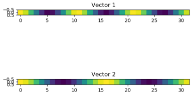


图 14: 两个基向量图像

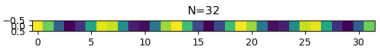


图 16: 两向量点乘过后的图像

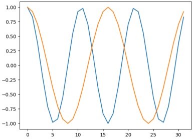


图 15: 两个基向量的曲线

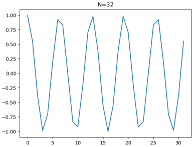


图 17: 两向量点乘过后的曲线

编程验证下正交性采样 $N = 128$ 时，基向量的图像

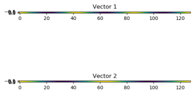


图 18: 两个基向量图像

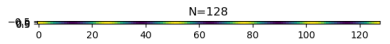


图 20: 两向量点乘过后的图像

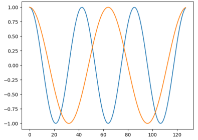


图 19: 两个基向量的曲线

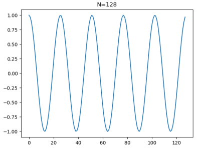


图 21: 两向量点乘过后的曲线

1-D DFT 的矩阵表达

$$F(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi k \frac{n}{N}}$$

$$F(k) = \begin{bmatrix} e^{-j2\pi k \frac{0}{N}} & e^{-j2\pi k \frac{1}{N}} \, e^{-j2\pi k \frac{2}{N}} & \dots & e^{-j2\pi k \frac{N-1}{N}} \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \mathbf{F(N-1)} \end{bmatrix}$$

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ \mathrm{F(N-1)} \end{bmatrix} = \begin{bmatrix} e^{-j2\pi(0)\frac{0}{N}} & e^{-j2\pi(0)\frac{1}{N}} & e^{-j2\pi(0)\frac{2}{N}} & & e^{-j2\pi(0)\frac{N-1}{N}} \\ e^{-j2\pi(1)\frac{0}{N}} & e^{-j2\pi(1)\frac{1}{N}} & e^{-j2\pi(1)\frac{2}{N}} & & e^{-j2\pi(1)\frac{N-1}{N}} \\ e^{-j2\pi(2)\frac{0}{N}} & e^{-j2\pi(2)\frac{1}{N}} & e^{-j2\pi(2)\frac{2}{N}} & \dots & e^{-j2\pi(2)\frac{N-1}{N}} \\ \mathrm{N-1} & & & & \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \mathbf{X(N-1)} \end{bmatrix}.$$

Q: 正变换矩阵 F 乘以逆变换矩阵 F^{-1} 等于什么？

1. 变换域基础

2. 离散傅里叶变换（Discrete Fourier Transform）
   - 一维离散傅里叶变换
   - 二维离散傅里叶变换
   - 数字图像的 DFT 应用

3. 离散余弦变换（Discrete Cosine Transform）

4. 参考文献

二维离散傅立叶变换的定义 [数 03]

$N*N$ 大小的图像 $x(n,m)$ 对应的二维离散傅里叶变换（2-D DFT）为:

$$
\begin{aligned}
F(k,l) &= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} x(n,m)W(n,m,k,l) \\
&= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} x(n,m)e^{-j2\pi\left(\frac{kn}{N}+\frac{lm}{N}\right)} \\
&= \sum_{n=0}^{N-1}\left[\sum_{m=0}^{N-1} x(n,m)e^{-j2\pi\frac{lm}{N}}\right]e^{-j2\pi\frac{kn}{N}}
\end{aligned}
$$

可以看成先对列做一维 DFT，再对行做一维 DFT。

二维离散傅里叶变换的性质

- 随着 k, l 的增加，频率也越来越大。
- $F(0,0)$ 代表着 DC（直流）成分，对应着像素的加权灰度值。

$$F(0,0) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(n,m)$$

- $F(k,l)$，当 $k > 0, l > 0$ 时，代表着 AC (交流) 成分。

二维离散傅里叶变换的基图像

$$F(k,l)=\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}x(n,m)e^{-j2\pi\left(\frac{kn}{N}+\frac{lm}{N}\right)}$$

$$f(x,y)=\frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}F(k,l)e^{j2\pi\left(\frac{kn}{N}+\frac{lm}{N}\right)}$$

- 跟一维 DFT 类似，我们看下二维 DFT 的“基向量”图像。
- 可以看出来二维 DFT 的“基向量”是 $N*N$ 的矩阵或者图像。
- 逆变换同样是“基向量”的线性叠加。
- 我们画出频域“基向量”的对偶空间图像。

二维 DFT 的对偶基图像

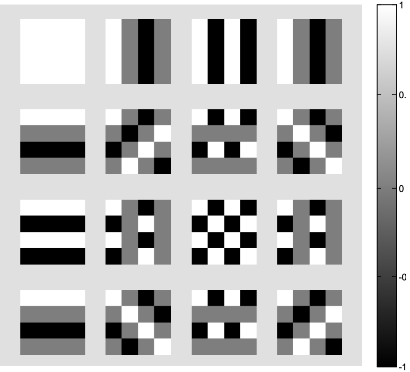


图 22: N=4 时，二维 DFT 对偶基 $e^{-j2\pi(\frac{kn}{N}+\frac{lm}{N})}$ 图像，原点是 DC 直流成分，右下角频率越大。

二维 DFT 的计算过程

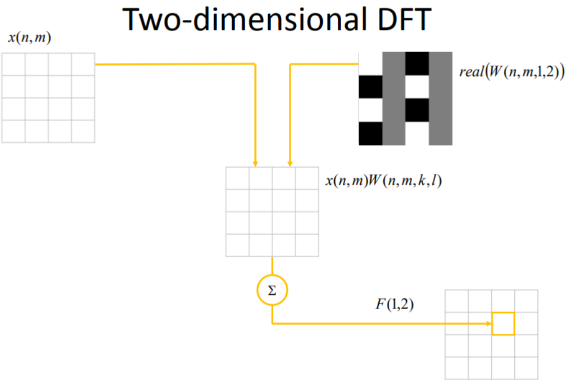


图 23: 二维 DFT 的计算过程

二维 DFT 的计算过程

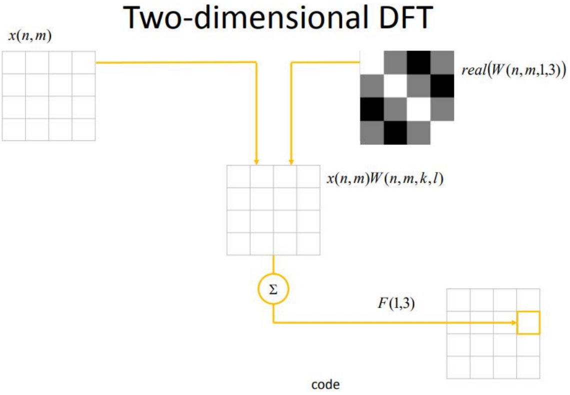


图 24: 二维 DFT 的计算过程

DFT 的性质

DFT 以及逆变换具有周期性，周期为 N。

$$F(k+N)=\sum_{n=0}^{N-1}x(n)e^{-j2\pi n\frac{k+N}{N}}$$
$$=\sum_{n=0}^{N-1}x(n)e^{-j2\pi n\frac{k}{N}}e^{-j2\pi n}=F(k)$$

$$x(n+N)=\frac{1}{N}\sum_{k=0}^{N-1}x(k)e^{j2\pi k\frac{n+N}{N}}=x(n)$$

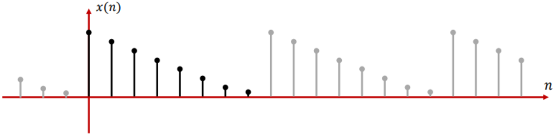


图 25: 周期性

DFT 的性质

DFT 具有共轭对称性（当两个复数，实部相同，虚部相反时）。

$$F(u,v) = F^*(-u,-v)$$

$$|F(u,v)| = |F(-u,-v)|$$

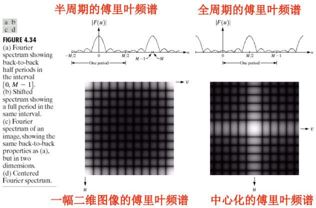
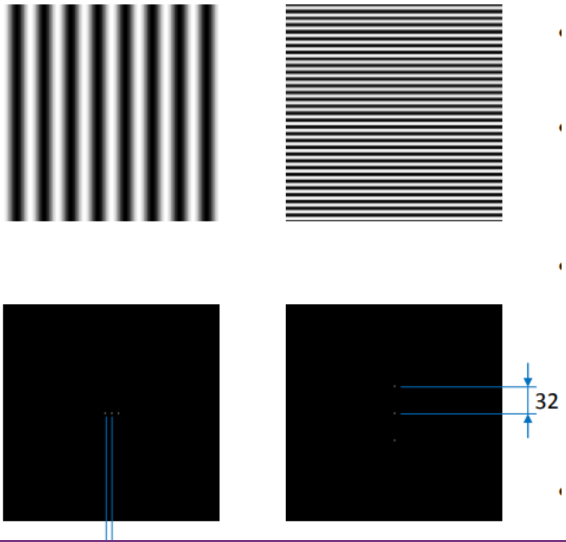
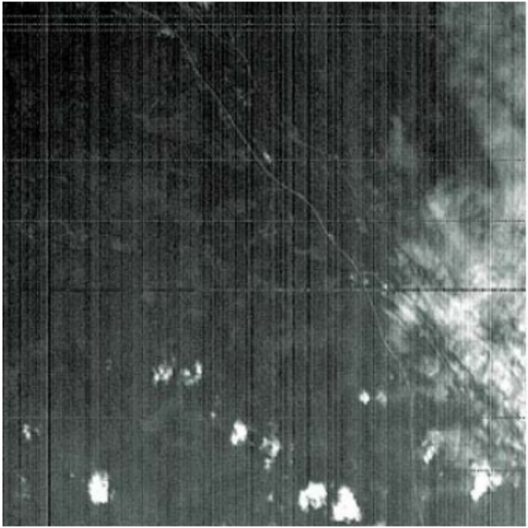


图 26: 频谱的中心化处理

数字图像的频谱



DFT 去噪举例



DFT 去噪举例

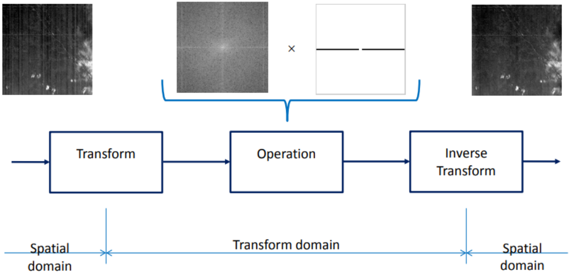


图 29: 频域去噪过程

- DFT 到频域
- 频域乘法，去除横向高频部分
- IDFT 逆变换到空间域，处理完成。

频域滤波器

频域滤波器的数学表达式如下：

$$G(u,v) = H(u,v)F(u,v)$$

- $F(u,v)$ 为图像的频域，$H(u,v)$ 为频域滤波函数，$G(u,v)$ 是处理过后的图像频域。
- 频域的乘法对应着空间域的卷积运算

$$g(x,y) = h(x,y) * f(x,y)$$

理想的低通滤波器

The two dimensional analogue of this is the function

$$H(u,v)=\begin{cases}1 & \text{if } \sqrt{u^2+v^2}\le w_0\\ 0 & \text{otherwise,}\end{cases}$$

where w_0 is now the cut-off frequency.

Thus, all frequencies inside a radius w_0 are kept, and all others discarded.

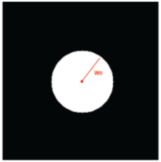


图 30: 理想的低通滤波器

离散余弦变换

- 因为离散余弦变换（DCT）具有更好的频谱能量聚集 (图像的重要信息集中在一起)，所以 DCT 常用在图像压缩领域，如 JPEG 格式采用了 DCT 变换进行图像压缩。
- 公式定义如下：

$$F(k)=\alpha(k)\sum_{n=0}^{N-1}x(n)\cos\left[\frac{(2n+1)k\pi}{2N}\right]$$

$$k=0,\ldots,N-1$$

$$\alpha(k)=\begin{cases}\sqrt{\dfrac{1}{N}} & ;\quad k=0\\[2ex] \sqrt{\dfrac{2}{N}} & ;\quad k\neq 0\end{cases}$$

This normalization parameters makes the DCT matrix orthonormal

图 31: 一维离散余弦变换

离散余弦变换

$$x(n) = \sum_{k=0}^{N-1} \alpha(k) F(k) \cos\left[\frac{(2n+1)k\pi}{2N}\right]$$

$$n = 0,\ldots,N-1$$

$$\alpha(0) = \sqrt{\frac{1}{N}}$$

$$\alpha(n) = \sqrt{\frac{2}{N}}$$

图 32: 一维离散余弦反变换

DCT 数学原理比较复杂，超出数字图像处理课程要求，感兴趣的同学可以查阅相关资料。

[数 03] 数字图像处理:.
国外优秀信息科学与技术系列教学用书. 电子工业出版社, 2003.

Thanks!