

## 中文摘要

随着科学技术的发展,我觉得所有工程问题的核心问题最终将归结为最优化问题(Optimization Problem)。有很多现成研究表明,不少图像处理问题也可转化为求线性系统的欠定方程组的最稀疏解。图像去噪问题也是其中之一。

本文围绕基于稀疏字典学习的图像去噪方法展开文献调研,综述了基于 DCT 过完备字典和 K-SVD 的图像去噪的具体原理,进一步的进行了实验与分析。

**关键词:** 稀疏表示; 图像去噪; DCT; K-SVD;

## 目 录

第 1 章	绪论 .....	1
1.1	研究背景 .....	1
1.2	本文主要内容 .....	1
第 2 章	稀疏表示以及稀疏分解算法 .....	2
2.1	稀疏表示 .....	2
2.2	欠定线性方程组的稀疏解 .....	3
2.2.1	匹配追踪算法(MP) .....	4
2.2.2	正交匹配追踪算法(OMP).....	5
2.3	本章小结 .....	8
第 3 章	基于字典学习稀疏表示到图像去噪 .....	9
3.1	基于 DCT 过完备字典的稀疏表示 .....	9
3.1.1	构造 DCT 过完备字典 .....	9
3.1.2	稀疏表示算法 .....	9
3.2	基于 K-SVD 的稀疏表示 .....	10
3.3	稀疏表示到图像去噪 .....	12
3.4	本章小结 .....	14
第 4 章	仿真实验与分析 .....	15
4.1	模型 .....	15
4.2	仿真设计及结果 .....	16
4.2.1	基于 DCT 过完备字典的稀疏表示 .....	16
4.2.2	基于 K-SVD 的稀疏表示.....	17
4.2.3	图像重构 .....	19
4.2.4	DCT 与 K-SVD 去噪性能分析.....	19
第 5 章	总结与感悟 .....	21
参考文献	.....	22

# 第1章 绪论

## 1.1 研究背景

我们生活在 21 世纪的信息化时代，科学研究和统计表明，人类从外界获得的信息约有 75%来自视觉系统，即图像。因为，实际生活中的图像均存在各种噪声，而且这些噪声会影响视觉效果等不利影响，因此对图像的去噪处理是一个重要研究领域。

稀疏表示已成为数据处理和信号分析的一种典范模式，所谓稀疏性是指：一个高维信号向量中只有少量的非 0 有效值或一个变换下仅有少量非 0 有效变换系数。其中 DCT 过完备字典和 K-SVD 是基于稀疏表示理论的一种有效的图像去噪方法。

## 1.2 本文主要内容

本文主要内容：

第一章，介绍稀疏表示和图像去噪的研究意义。

第二章，介绍稀疏表示理论和线性方程组的稀疏解问题以及详细描述两种贪心算法的具体原理。

第三章，与第二章所介绍的稀疏解问题跟图像去噪问题相结合，基于 DCT 过完备字典和 K-SVD 的方法稀疏表示解决图像去噪问题。

第四章，进行稀疏表示和图像去噪的仿真实验以及分析。

第五章，总结与感悟。

## 第2章 稀疏表示以及稀疏分解算法

### 2.1 稀疏表示

在优化目标问题中，目标函数为凸函数，则问题有唯一解。在信号的稀疏表示时，不同范数会起到不同作用。对  $N$  维向量， $l_p$  范数定义为：

$$\|\mathbf{x}\|_p^p = \left( \sum_{i=1}^p |x_i|^p \right)^{\frac{1}{p}}$$

定义范数的四个基本性质保证了当  $p \geq 1$  时， $l_p$  范数为凸函数的范数，即  $l_1$  范数为同时满足范数定义与凸函数性的最小  $p$  值。因为当  $0 < p < 1$  时， $l_p$  范数不满足三角不等式，并且此时该范数为非凸函数。当  $p = 0$  时，即  $l_0$  范数不满足范数的齐次性，它也是一个非凸函数。因此，严格来讲当  $p < 1$  时， $l_p$  范数不再是一个真范数，但它的性质确实比较接近范数概念，所以只是借用了这个名词。

若优化问题中，目标函数取  $J(\mathbf{x}) = \|\mathbf{x}\|_p^p$ 。

则当  $p > 1$  时， $\mathbf{x}$  中绝对值大的分量会对目标函数的影响较大，而绝对值小的分量应将较小，是因为它们会  $p$  次方贡献。换句话说， $p > 1$  的  $l_p$  范数，对小分量的权重相对小，容易被忽视，则导致最终优化结果也很难保证  $\mathbf{x}$  的稀疏化。

相反的，当  $p = 0$  时，由于  $l_0$  范数只考虑非零元素的个数，而不考虑各个分量的大小，因此它很容易达到  $\mathbf{x}$  的稀疏化。但是  $l_0$  范数前面提到，它是一个非凸的，因此本身很难优化。同样的，只要  $p < 1$ ， $l_p$  范数拥有优化上的困难。

当  $p = 1$ ，即  $l_1$  范数克服了上面所提到的两个问题。因为它即不会起到每个分量值的大小上的变化，也相对来说很容易优化。所以  $l_1$  范数是唯一具有稀疏化能力的凸函数的范数。

对于一个长度为  $n$  的信号向量，所谓“稀疏表示”是指：可以利用  $K(K \ll n)$  个非零系数在某个框架下准确表示。此时  $K$  称为稀疏度。严格地描述：一个大多数零元素的向量或者矩阵称为稀疏向量或稀疏矩阵。若向量  $\mathbf{y} \in R^m$  可分解为  $n$  个  $m$  维向量  $\mathbf{a}_i \in R^m, i = 1, \dots, n$  (其中  $n > m$ ) 的线性组合<sup>[6]</sup>：

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \sum_{i=1}^n x_i \mathbf{a}_i \quad (n > m) \quad (2-1)$$

则  $n$  个向量  $\mathbf{a}_i \in R^m, i = 1, \dots, n$ ,  $n$  不可能是正交基的集合。为了与基区分, 这些列向量通常被称为原子(Atom)。由于原子的个数  $n$  大于向量空间  $R^m$  的维数, 所以称这些原子的集合是过完备(Over-complete)的。过完备的原子组成的矩阵  $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in R^{m \times n} (n > m)$  称为字典(Dictionary)。

由于式 (2-1) 为欠定方程, 因此有无穷多组解。为了求解这个方程, 需要施加约束条件, 如最小  $l_2$  范数解或稀疏解等。

当向量  $\mathbf{x}$  为稀疏向量时, 信号分解  $\mathbf{y} = \mathbf{Ax}$  称为稀疏分解(Sparse decomposition), 或称目标信号  $\mathbf{y}$  相对于字典  $\mathbf{A}$  的稀疏表示(Sparse representation)。

## 2.2 欠定线性方程组的稀疏解

设一个满秩矩阵  $\mathbf{A} \in R^{m \times n} (n > m)$ , 定义式 (2-1) 为欠定线性方程组。定义一般优化问题  $P_J$ :

$$(P_J): \quad \min_{\mathbf{x}} J(\mathbf{x}) \quad \text{subject to} \quad \mathbf{b} = \mathbf{Ax}. \quad (2-2)$$

选择一个严格凸函数  $J(\cdot)$  保证唯一解。问题  $P_0$  中选择  $J(\mathbf{x}) = \|\mathbf{x}\|_0$ , 即:

$$(P_0): \quad \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{b} = \mathbf{Ax}. \quad (2-3)$$

但是, 因为  $l_0$  范数是一个非凸的, 存在求解上的困难。这个问题一个是 NP-Hard 问题<sup>[5]</sup>。一般解决此问题的方法是对其做松弛或用贪心算法得到近似解。

其中, 凸松弛(Convex relaxation)算法就是用一个凸函数  $J(\mathbf{x})$  来替代  $\|\mathbf{x}\|_0$ , 从而将原非凸优化问题转化为可以用标准算法求解的凸优化问题。一种自然的方法是用  $l_p (p > 0)$  范数来替代  $l_0$  范数。如前所述,  $\|\mathbf{x}\|_1$  是对原目标函数  $\|\mathbf{x}\|_0$  的最佳凸松弛近似。此时, 稀疏性约束下的欠定方程求解非凸问题  $P_0$  近似为保持稀疏性的一种优化问题  $P_1$ <sup>[5]</sup>:

$$(P_1): \quad \min_{\mathbf{x}} \|\mathbf{Wx}\|_1 \quad \text{subject to} \quad \mathbf{b} = \mathbf{Ax}. \quad (2-4)$$

其中, 矩阵  $\mathbf{W}$  是引入补偿权值的对角正定矩阵。在这种情况下, 矩阵中  $(i, i)$  项的自然选择是  $w(i) = \|\mathbf{a}_i\|_2$ 。当  $\mathbf{A}$  的所有列归一化后, 即  $\mathbf{W}=\mathbf{I}$ , 该模型称为基追踪(Basic Pursuit)。其  $P_1$  问题可通过线性规划来解决。

但在实际中，采用凸优化算法解决这种问题的运算复杂度很高。因此咱可以选择运算复杂度更低的方法，其中贪心算法是很简单，同时有效的方法之一。近似问题中，最基本的贪心算法为匹配追踪算法以及和对其做改进的正交匹配追踪算法。

### 2.2.1 匹配追踪算法(MP)

假定被表示的信号为  $\mathbf{y} \in R^m$ ，假定  $\mathbf{H}$  表示 Hilbert 空间，在这个空间  $\mathbf{H}$  里，由一组向量  $\{\mathbf{x}_i\}$  ( $i = 1, \dots, n$ ) 构成字典矩阵  $\mathbf{D}$ ，而且这些向量已经被归一化的，即  $\|\mathbf{x}_i\| = 1$ 。（这里归一化的原因是它会被用到计算各个分量的内积投影长度，若某个分量相对很大，会导致生成次优解）

匹配追踪(Matching Pursuit)算法是在这字典矩阵  $\mathbf{D}$  中，选择一个跟目标信号  $\mathbf{y}$  最匹配的原子（向量在空间  $\mathbf{H}$  中定义，因此匹配程度由两向量的内积来决定，即寻找一个内积最大的原子），并求出信号的残差，然后继续同样选择与这残差最匹配的原子，直到残差值可以忽略，则由这些原子的线性组合表示  $\mathbf{y}$  即完成稀疏分解。其流程图<sup>[7]</sup>：

- (I) Compute the inner-products  $\{\langle \mathbf{R}_k f, x_n \rangle\}_n$ .
- (II) Find  $n_{k+1}$  such that
- $$|\langle \mathbf{R}_k f, x_{n_{k+1}} \rangle| \geq \alpha \sup_j |\langle \mathbf{R}_k f, x_j \rangle|,$$
- where  $0 < \alpha \leq 1$ .
- (III) Set,
- $$\begin{aligned} f_{k+1} &= f_k + \langle \mathbf{R}_k f, x_{n_{k+1}} \rangle x_{n_{k+1}} \\ \mathbf{R}_{k+1} f &= \mathbf{R}_k f - \langle \mathbf{R}_k f, x_{n_{k+1}} \rangle x_{n_{k+1}} \end{aligned}$$
- (IV) Increment  $k$ , ( $k \leftarrow k+1$ ), and repeat steps (I)–(IV) until some convergence criterion has been satisfied.

图 2-1 MP 算法流程图

其中， $\mathbf{R}_k f$  表示信号的残差。

容易想到，MP 算法有严重的缺陷。它只能保证  $\mathbf{R}_k f \perp \mathbf{x}_k$ ，而不能保证  $\mathbf{R}_k f \perp V_k$ ，其中  $V_k = \overline{\text{span}(V_1, \dots, V_{k-1})}$ 。即如果残差在原子上的垂直投影不满足正交性

的话，这会导致迭代结果是次优而不是最优的。因此引入正交匹配追踪(Orthogonal Matching Pursuit)算法。

### 2.2.2 正交匹配追踪算法(OMP)

正交匹配追踪(Orthogonal Matching Pursuit)算法，即 OMP 算法与 MP 算法区别在于，它在稀疏分解的每一步对当前以及之前所选择的原子做正交化处理。其流程图<sup>[7]</sup>：

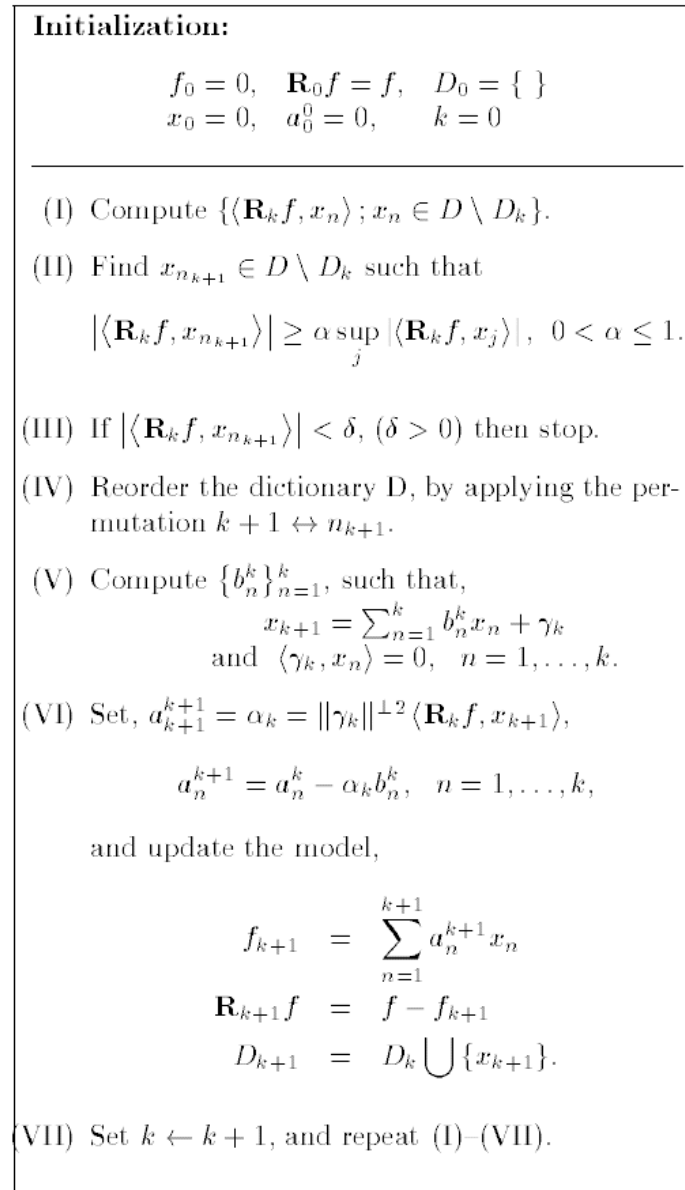


图 2-2 OMP 算法流程图

我们的目的是通过贪心算法求问题 $P_0$ 即式（2-3）的近似解。OMP 算法中，

$$f = \sum_{n=1}^k a_n^k x_n + R_k f, \text{ 其中 } \langle R_k f, x_n \rangle = 0 \ (n = 1, \dots, k)$$

则设  $f$  在  $A$  上的正交投影为  $P=Ax$ 。令  $r = f - Ax$ ，则  $r$  垂直于  $A$ ，如下图：

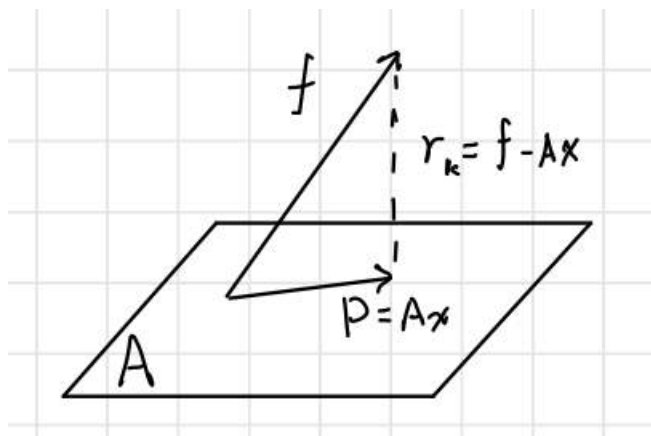


图 2-3 Schimdt 正交化

由  $r_k \perp A$  得：

$$\langle A, r_k \rangle = A^T (f - Ax) = 0$$

其解为：

$$x_{tls} = (A^T A)^{-1} A^T f \quad (2-5)$$

此时，

$$P = Ax_{tls} = A(A^T A)^{-1} A^T f \quad (2-6)$$

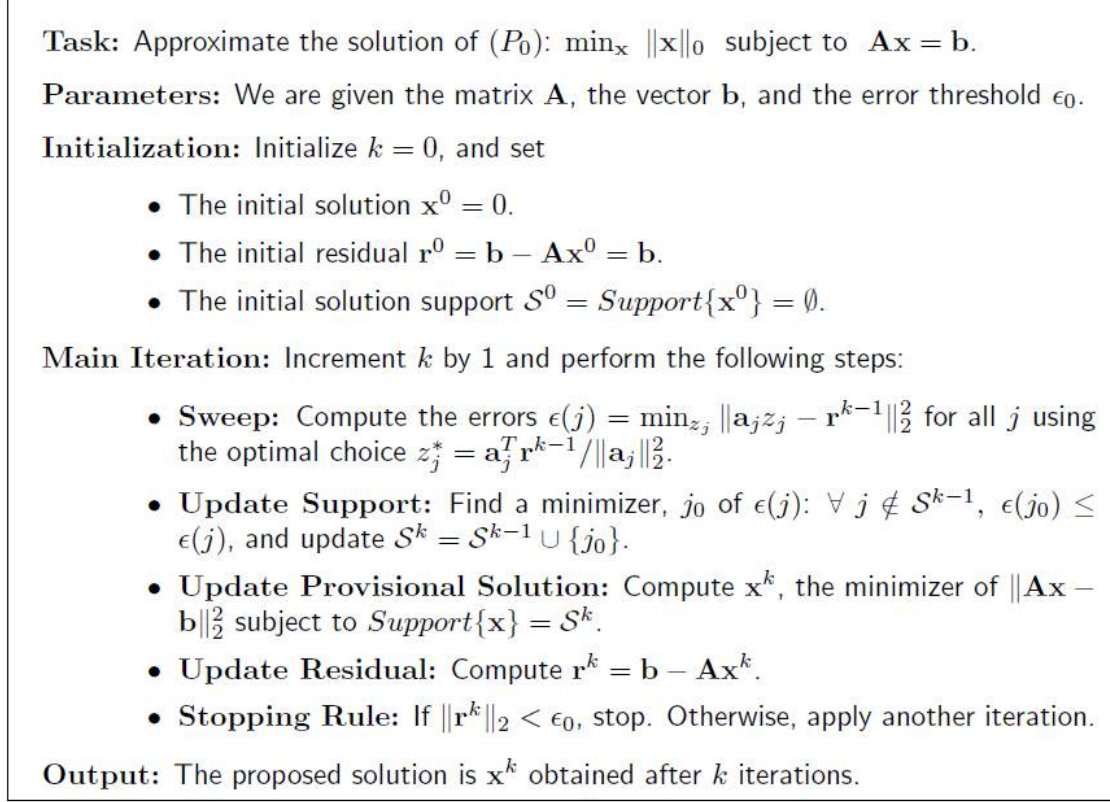
因此，

$$r_k = f - Ax_{tls} = f - A(A^T A)^{-1} A^T f \quad (2-7)$$

由上述的 Schimdt 正交化与 MP 算法结合可以得到近似解。



在文献[5]中其流程图为<sup>[5]</sup>:



0

图 2-4 OMP 算法求解 $P_0$

其实，这原理跟图 2-2 没区别，图 2-3 中，Sweep 阶段就是在寻找最匹配的原子在字典  $\mathbf{A}$  中的列索引。然后在 Update Provisional Solution 阶段， $\mathbf{x}$  的最小  $l_2$  范数解可通过  $\mathbf{A}$  的伪逆乘以  $\mathbf{b}$  获得，即<sup>[6]</sup>:

$$\hat{\mathbf{x}} = \mathbf{A}^+ \mathbf{b} = \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{b} \quad (2-8)$$

## 2.3 本章小结

本章首先给出稀疏表示有关的基本概念，然后引入求解线性方程组的最稀疏解问题。稀疏分解问题，即欠定线性方程组的最小 $l_0$ 范数解问题是一个 **NP-Hard** 问题，不容易直接求解，因此有两种算法来求该问题的近似解：贪心算法和凸松弛算法，其中最基本的贪心算法可分为 **MP** 算法和对其做改进的 **OMP** 算法。本章详细介绍了对这两种算法的基本原理。这些算法原理将作为第 3 章的基础知识。

## 第3章 基于字典学习稀疏表示到图像去噪

### 3.1 基于 DCT 过完备字典的稀疏表示

#### 3.1.1 构造 DCT 过完备字典

一维离散余弦变换(DCT)为:

$$F(x, u) = \alpha(u) \sum_{n=0}^{N-1} f(x) \cos \frac{(2x+1)n\pi}{2N} \quad (3-1)$$

其中,

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{M}} & , u = 0 \\ \sqrt{\frac{2}{M}} & , u \neq 0 \end{cases}$$

$u, x = 1, \dots, N-1$ 。

#### 3.1.2 稀疏表示算法

为了方便分析,我们先对局部图像块进行分析,在将推广到全局图像。

首先我们考虑从大小为  $M \times N$  的原始图像  $\mathbf{X}$  中取出大小为  $m \times n$  的图像块,然后将这些图像块排成一个列向量  $\mathbf{x}$ 。

此时,我们定义一个冗余字典  $\mathbf{D}$  来构造稀疏表示模型:

$$\hat{\alpha} = \arg \min_{\alpha} \|\alpha\|_0 \text{ subject to } \mathbf{D}\alpha \approx \mathbf{x} \quad (3-2)$$

模型中,为了保证稀疏,  $\|\hat{\alpha}\|_0 \leq K < mn$ , 并且需要明确的表示误差标准即  $\|\mathbf{D}\alpha - \mathbf{x}\|_2 \leq \varepsilon$ 。从这个模型可以看出,每个图像块都可以由  $\mathbf{D}$  的少量原子的线性组合来表示。

如果我们在原始图像块加入了标准差为  $\sigma$  的零均值高斯白噪声,则得到噪声图像  $\mathbf{y}$ 。由于是高斯噪声,重构误差我们采用  $l_2$  范数对其测度,则问题变为:

$$\hat{\alpha} = \arg \min_{\alpha} \|\alpha\|_0 \text{ subject to } \|\mathbf{D}\alpha - \mathbf{y}\|_2^2 \leq T \quad (3-3)$$

其中  $T$  由表示误差  $\varepsilon$  和噪声方差  $\sigma$  来决定。并且上式又可表示为：

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{D}\alpha - \mathbf{y}\|_2^2 + \mu \|\alpha\|_0 \quad (3-4)$$

显然此问题可以通过 OMP 算法很容易得到近似解。

### 3.2 基于 K-SVD 的稀疏表示

将上述局部模型推广到全局模型，则问题变为：

$$\min_{X, D} \|\mathbf{D}\mathbf{X} - \mathbf{Y}\|_2^2 \text{ subject to } \|x_i\|_0 \leq K \quad (3-5)$$

其中， $K$  为稀疏度。

K-SVD 的基本思想是交替最小化，即：

第一步，先固定  $\mathbf{D}_{k-1}$ ，找出最稀疏的  $\mathbf{X}_k$  使得  $\mathbf{Y} = \mathbf{D}_{k-1}\mathbf{X}_k$ 。

第二步，再固定  $\mathbf{X}_k$ ，找出  $\mathbf{D}_k$  使得  $\mathbf{D}_k = \min_D \|\mathbf{Y} - \mathbf{D}\mathbf{X}_k\|_F^2$ 。

如此重复，即可得到优化字典  $\mathbf{D}$ 。

其中第一步同样可以由 OMP 算法得到；而第二步，

$$\begin{aligned} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 &= \|\mathbf{Y} - \sum_{i=1}^K d_i x_i^T\|_F^2 \\ &= \|\mathbf{Y} - \sum_{i=1}^K d_i x_i^T - d_k x_k^T\|_F^2 \\ &\triangleq \|\mathbf{E}_k - d_k x_k^T\|_F^2 \end{aligned} \quad (3-6)$$

上述方程可以用求解最小二乘方法求解 $d_k$ ，但此 $d_k$ 不是单位向量。因此我们需要用奇异值分解（SVD）。但还不能直接对 $E_k$ 进行 SVD，因为这样会导致 $x_k$ 不稀疏。所以先要把 $x_k$ 只保留非零项 $x_{k_R}$ ，接着 $E_k$ 也只保留 $x_{k_R}$ 对应的项 $E_{k_R}$ 。然后再对 $E_{k_R}$ 进行 SVD，更新 $x_k$ 和 $d_k$ 。文献[1]中的 K-SVD 算法流程图为：

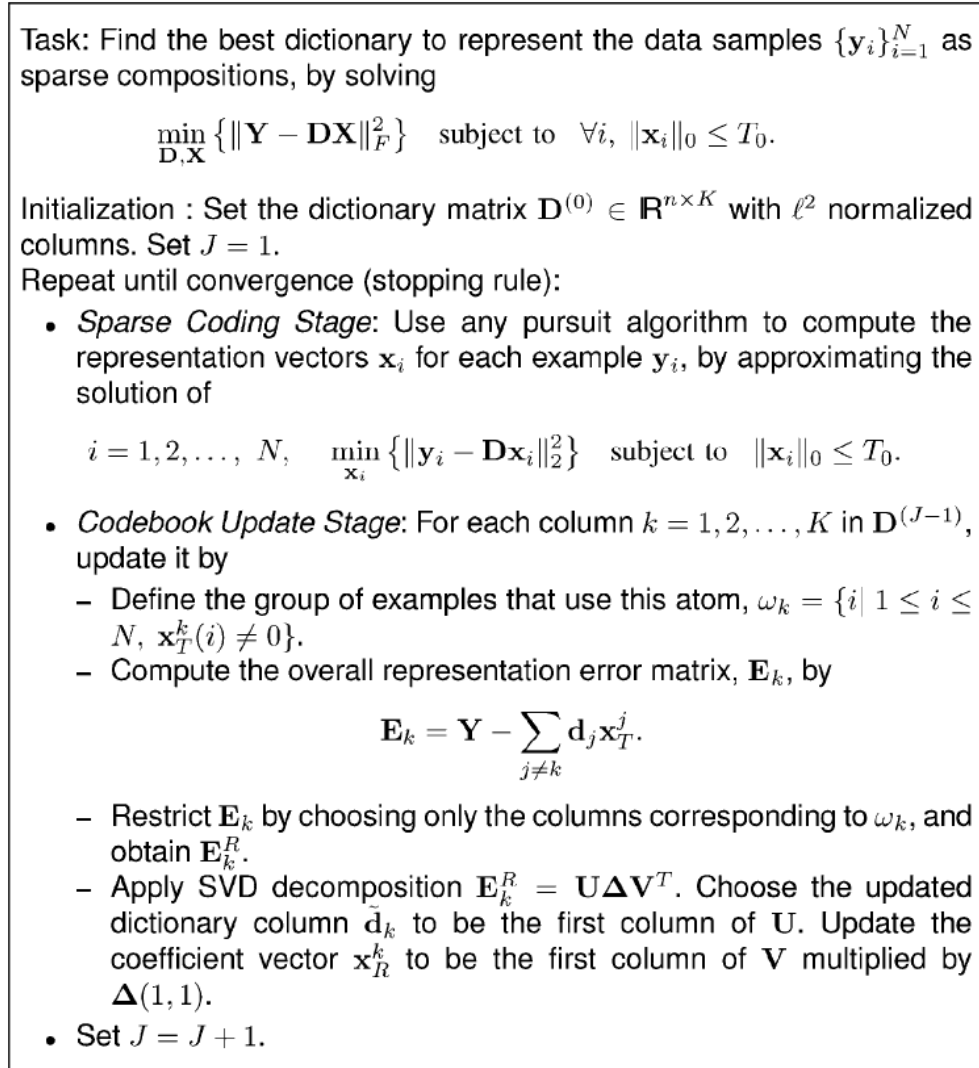


图 3-1 K-SVD 算法

### 3.3 稀疏表示到图像去噪

去噪问题相当于是一个带约束条件（稀疏性）的最小化问题。前面式（3-4）已给出噪声局部图像的稀疏表示模型。此时我们可通过  $\hat{\mathbf{x}} = \mathbf{D}\hat{\boldsymbol{\alpha}}$  可获得图像块  $\mathbf{x}$  的局部去噪模型。

则推广到全局图像，该模型变为：

$$\{\hat{\boldsymbol{\alpha}}_{ij}, \hat{\mathbf{X}}\} = \arg \min_{\boldsymbol{\alpha}_{ij}, \mathbf{X}} \lambda \|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_{i,j} \mu_{ij} \|\boldsymbol{\alpha}_{ij}\|_0 + \sum_{i,j} \|\mathbf{D}\boldsymbol{\alpha}_{ij} - \mathbf{R}_{ij}\mathbf{X}\|_2^2 \quad (3-7)$$

其中，第一项表示噪声图像  $\mathbf{Y}$  和去噪图像  $\mathbf{X}$  要相接近，并且其约束取决于  $\lambda$ ，它是一个与  $\sigma$  有关的常数。而第二项和第三项是图像先验，它们说明每个图像块  $\mathbf{x}_{ij} = \mathbf{R}_{ij}\mathbf{X}$  在有界误差内具有稀疏表示。

进一步的，式（3-7）可以分为：

$$\hat{\boldsymbol{\alpha}}_{ij} = \arg \min_{\boldsymbol{\alpha}_{ij}} \sum_{i,j} \mu_{ij} \|\boldsymbol{\alpha}_{ij}\|_0 + \|\mathbf{D}\boldsymbol{\alpha}_{ij} - \mathbf{x}_{ij}\|_2^2 \quad (3-8)$$

$$\hat{\mathbf{X}} = \arg \min_{\mathbf{X}} \lambda \|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_{i,j} \|\mathbf{D}\hat{\boldsymbol{\alpha}}_{ij} - \mathbf{R}_{ij}\mathbf{X}\|_2^2 \quad (3-9)$$

我们从  $\mathbf{X}=\mathbf{Y}$  开始，则式（3-8）可对每个图像块进行 OMP 算法而得到稀疏解。此时，为了求解式（3-9）的方程，我们可以构造能量函数：

$$g(\mathbf{X}) = \lambda \|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_{i,j} \|\mathbf{D}\hat{\boldsymbol{\alpha}}_{ij} - \mathbf{R}_{ij}\mathbf{X}\|_2^2 \quad (3-9)$$

并且令

$$\begin{aligned} \frac{\partial g(\mathbf{X})}{\partial \mathbf{X}} &= \frac{\partial}{\partial \mathbf{X}} \left[ \lambda \|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_{i,j} \|\mathbf{D}\hat{\boldsymbol{\alpha}}_{ij} - \mathbf{R}_{ij}\mathbf{X}\|_2^2 \right] \\ &= 2\lambda(\mathbf{X} - \mathbf{Y}) + \sum_i 2(-\mathbf{R}_{ij})(\mathbf{D}\hat{\boldsymbol{\alpha}}_{ij} - \mathbf{R}_{ij}\mathbf{X}) \\ &= 2 \left( \lambda \mathbf{I} + \sum_i \mathbf{R}_{ij}^T \mathbf{R}_{ij} \right) \mathbf{X} - 2 \left( \lambda \mathbf{Y} + \sum_i \mathbf{R}_{ij}^T \mathbf{D}\hat{\boldsymbol{\alpha}}_{ij} \right) = 0 \end{aligned}$$

从而得到：

$$\hat{\mathbf{X}} = \left( \lambda \mathbf{I} + \sum_i \mathbf{R}_{ij}^T \mathbf{R}_{ij} \right)^{-1} \left( \lambda \mathbf{Y} + \sum_i \mathbf{R}_{ij}^T \mathbf{D} \hat{\alpha}_{ij} \right) \quad (3-10)$$

其实，这个式子只是说明要对去噪的图像块进行平均，可通过对噪声图像块进行平均而得到。

下面过程就是基于 K-SVD 的稀疏表示去噪的方法。流程图如下<sup>[3]</sup>：

**Algorithm Parameters:**  $n$  - block size,  $k$  - dictionary size,  $J$  - number of training iterations,  $\lambda$  - Lagrange multiplier, and  $C$  - noise gain.

$$\min_{\mathbf{X}, \mathbf{D}, \mathbf{A}} \left\{ \lambda \|\mathbf{Y} - \mathbf{X}\| + \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_0 + \sum_{ij} \|\mathbf{D} \alpha_{ij} - R_{ij} X\|_2^2 \right\}$$

1. Initialization : Set  $\mathbf{X} = \mathbf{Y}$ ,  $\mathbf{D}$  = overcomplete DCT dictionary.

2. Repeat  $J$  times:

- *Sparse Coding Stage:* Use any pursuit algorithm to compute the representation vectors  $\alpha_{ij}$  for each patch  $R_{ij}\mathbf{X}$ , by approximating the solution of

$$\forall_{ij} \min_{\alpha_{ij}} \|\alpha_{ij}\|_0 \quad \text{s.t.} \quad \|R_{ij}\mathbf{X} - \mathbf{D} \alpha_{ij}\|_2^2 \leq (C\sigma)^2.$$

- *Dictionary Update Stage:* For each column  $l = 1, 2, \dots, k$  in  $\mathbf{D}$ , update it by

– Find the set of patches that use this atom,  $\omega_l = \{(i, j) | \alpha_{ij}(l) \neq 0\}$ .

– For each index  $(i, j) \in \omega_l$ , compute its representation error

$$\mathbf{e}_{ij}^l = R_{ij}\mathbf{X}_{ij} - \sum_{m \neq l} \mathbf{d}_m \alpha_{ij}(m).$$

– set  $\mathbf{E}_l$  as the matrix whose columns are  $\{\mathbf{e}_{ij}^l\}_{(i,j) \in \omega_l}$

– Apply SVD decomposition  $\mathbf{E}_l = \mathbf{U} \Delta \mathbf{V}^T$ . Choose the updated dictionary column  $\bar{\mathbf{d}}_l$  to be the first column of  $\mathbf{U}$ . Update the coefficient values  $\{\alpha_{ij}(l)\}_{(i,j) \in \omega_l}$  to be the entries of  $\mathbf{V}$  multiplied by  $\Delta(1, 1)$ .

3. Set:

$$\mathbf{X} = \left( \lambda \mathbf{I} + \sum_{ij} R_{ij}^T R_{ij} \right)^{-1} \left( \lambda \mathbf{Y} + \sum_{ij} R_{ij}^T \mathbf{D} \alpha_{ij} \right)$$

图 3-2 基于稀疏表示图像去噪

### 3.4 本章小结

本章的核心问题是如何对图像进行稀疏表示并且去除图像的噪声。稀疏和冗余表示作为一种利用局部信息的模型。图像的去噪问题相当于是一个带约束条件（稀疏性）的最优化问题，因此本章中用到了在第二章所提到的求最稀疏解的算法，OMP 算法。稀疏表示以及去噪的第一种方法是固定冗余字典  $D$ ，通过 OMP 得到最稀疏解和去噪图像。第二种方法是利用 K-SVD 算法，它通过同时更新字典原子和已知稀疏向量，得到了一个有效描述图像信息的字典，从而同样可以得到去噪图像。



## 第4章 仿真实验与分析

### 4.1 模型

题目要求：

本作业中我们研究 K-SVD 在图像压缩与去噪中的应用[3]-[5]。给定一幅大小为  $M \times N$  的图像  $S$ （如  $512 \times 512$  的 Lena 图像），对其添加标准差为  $\sigma = 5$  的高斯白噪声。可以将含噪图像  $\hat{S}$  分为  $(M - n + 1) \times (N - n + 1)$  个大小为  $n \times n = 8 \times 8$  的滑动图像块。完成如下要求：

- (a) 将所有滑动图像块在  $64 \times 256$  的 DCT 字典上进行稀疏表示，计算参与表示的原子的平均个数。
- (b) 利用 K-SVD 算法对加噪图像本身进行字典的学习。当迭代次数从 1-20 时，统计进行稀疏表示的原子的平均个数，并与 (a) 的结果进行比较。
- (c) 设含噪图像经稀疏表示后重构的图像为  $\hat{S}$ ，则其去噪性能可通过

$$\text{PSNR} = 10 \log_{10} \left( \frac{M \times N}{\sum_{i,j} (s_{i,j} - \hat{s}_{i,j})^2} \right)$$

来衡量。试计算利用 DCT 字典及 K-SVD 学习字典进行图像恢复的 PSNR 值，并进行比较。

本实验中，选取了： $512 \times 512$  的 Lena 图像。

特别说明：为了研究参与稀疏表示的原子平均个数，对 OMP 算法的停止条件加强了（不然，它的稀疏度太低），但这样会导致计算量的增加（收敛慢），因此研究此问题时选取了 Lena 图像的前  $64 \times 64$  个像素。另外，研究去噪性能的时候，由于题目中给出的噪声标准差  $\sigma$  本身较小（如下图），因此为了直观观察去噪性能，研究问题(c)时  $\sigma$  暂定为 25。

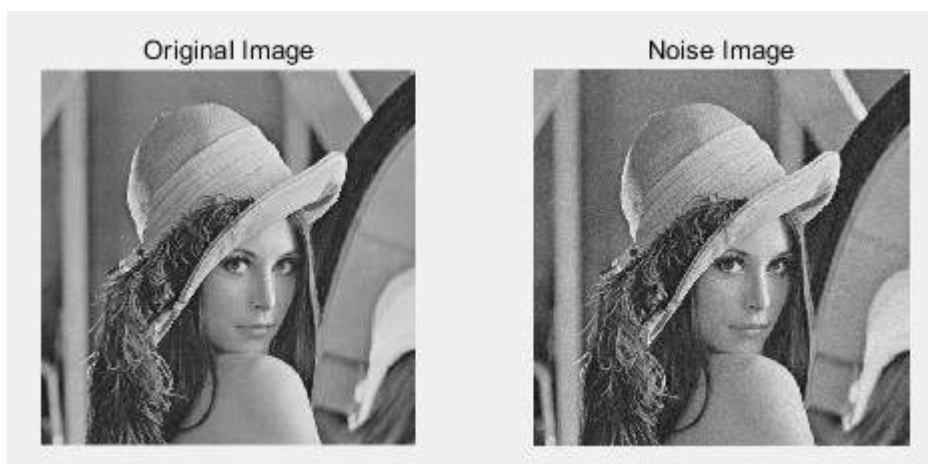


图 4-1  $\sigma = 5$  时的噪声图像

## 4.2 仿真设计及结果

### 4.2.1 基于 DCT 过完备字典的稀疏表示

核心代码:

```
%% DCT Dictionary
Dm=8;Dn=16;
for i=0:Dm-1
    for j=0:Dn-1
        if i==0
            C=sqrt(1/Dm);
        else
            C=sqrt(2/Dm);
        end
        A(i+1,j+1)=C*cos((2*j+1)*pi*i/2/Dm);
    end
end

DCT=kron(A,A);

%% OMP sparse coding
[Dm,Dn]=size(DCT);
for i=1:Dn    %% Normalization
    s=sum(DCT(:,i).^2);
    DCT(:,i)=DCT(:,i)./s;
end

[~,num]=size(Patch);
del=1e-3;
x=zeros(Dn,num);
for i=1:num    %% OMP
    Patch_in=Patch(:,i)-repmat(mean(Patch(:,i)),Dm,1);
    Ak=zeros(Dm,Dn);
    f=Patch_in;
    Rkf=f;
    for k=1:Dn
        [~,ind_k]=max(abs(DCT'*Rkf));
        index(k)=ind_k;
        Ak(:,k)=DCT(:,ind_k);
        x_ls=(Ak(:,1:k)'+Ak(:,1:k))\Ak(:,1:k)'+f;
        Rkf=f-Ak(:,1:k)*x_ls;
        % if max(abs(DCT'*Rkf))<del
        % break
        % end
        if sqrt(sum(Rkf.^2))<T
            break
        end
    end
    x(index(1:k),i)=x_ls;
    K(i)=length(x(x(:,i)>0));
    Patch_out(:,i)=DCT*x(:,i)+ones(Dm,1)*mean(Patch(:,i));
```

**说明及数值结果：**根据 3.1 节，首先构造  $64 \times 256$  的 DCT 字典。然后为了进行 OMP 算法，先把 DCT 字典按列归一化。OMP 算法停止条件 T 取了  $(C\sigma)^2$ ，即当  $\|D\alpha - y\|_2^2 \leq (C\sigma)^2$  时，停止。其中  $C=1.15$ 。

此时，参与稀疏表示的原子平均个数为：

Atom\_meanDCT =

4.5124

则相当于它今用了  $\frac{4.5124}{8 \times 8} = 7.05\%$  的数据表达了每个图像块。

若 T 取更小，即  $T = C\sigma$ ，此时参与稀疏表示的原子平均个数为：

Atom\_meanDCT =

22.9177

即约使用了 36% 的数据用来表达图像块。可以发现，稀疏表示时 T 取越小，则表示的精度确实会提高，可是非零项约多，这样不再是稀疏表示了。而且这样的话，会导致去噪性能不良（不能去除小分量的噪声）。

#### 4.2.2 基于 K-SVD 的稀疏表示

核心代码：

```
%K SVD Algorithm
KSVD=DCT;           %Initialization
[Km,Kn]=size(KSVD);
J=20;
for repeat=1:J
```

```

%% KSVD
V=Patch_in;dx=0;
for k=1:Kn
    Ek=V-KSVD*x+KSVD(:,k)*x(k,:);
    index=find(x(k,:)>0);
    if ~isempty(index)
        EkR=Ek(:,index);
        [U,S,V]=svd(EkR);
        KSVD(:,k)=U(:,1);
        x(k,index)=V(:,1)*S(1,1);
    end
end
for i=1:Kn
    K(i)=length(x(x(:,i)>0));
    Patch_out(:,i)=KSVD*x(:,i)+ones(Km,1)*mean(Patch(:,i));
end
K_KSVD(repeat)=mean(K);

```

**说明及数值结果：**根据 3.2 节所述的原理，得到基于 K-SVD 算法的稀疏表示。其中，OMP 算法停止条件 T 同样取 $(C\sigma)^2$ ，即当 $\|\mathbf{D}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 \leq (C\sigma)^2$  时，停止。其中 C=1.15。

当迭代 20 次，每次参与稀疏表示的原子平均个数为：

J	1	2	3	4	5	6	7
平均原子	4.3485	4.2860	4.2918	4.2758	4.2723	4.2835	4.2710
J	8	9	10	11	12	13	14
平均原子	4.2813	4.2787	4.2800	4.2816	4.2761	4.2851	4.2663
J	15	16	17	18	19	20	
平均原子	4.2758	4.2758	4.2825	4.2704	4.2666	4.2768	

此总平均原子数为：

$$\text{Atom\_meanKSVD} = 4.2813$$

则相当于它今用了  $\frac{4.2813}{8 \times 8} = 6.69\%$  的数据表达了每个图像块。可知其方法比 DCT 字典稀疏表示的方法拥有更好的稀疏表示特性。这符合理论分析。

### 4.2.3 图像重构

核心代码:

```
%% Image reconstruction
rltrl=zeros(M,N);
rltda=zeros(M,N);
[indr,indc]=ind2sub([M-n+1,N-n+1],1:num);
for i=1:(M-n+1)*(N-n+1)
    rltrl(indr(i):indr(i)+n-1,indc(i):indc(i)+n-1)=rltrl(indr(i):indr(i)+n-1,indc(i):indc(i)+n-1)+ones(n,n);
    rltda(indr(i):indr(i)+n-1,indc(i):indc(i)+n-1)=rltda(indr(i):indr(i)+n-1,indc(i):indc(i)+n-1)+reshape(Patch_dn(:,i),[8,8]);
end
X_DCT=(lambda+1_noise+rltda)./(lambda+ones(M,N)+rltrl);
```

**说明及数值结果:** 根据 3.3 节所述的原理, 通过局部去噪图像块的重构可以得到全局去噪图像。其效果将在下面验证。

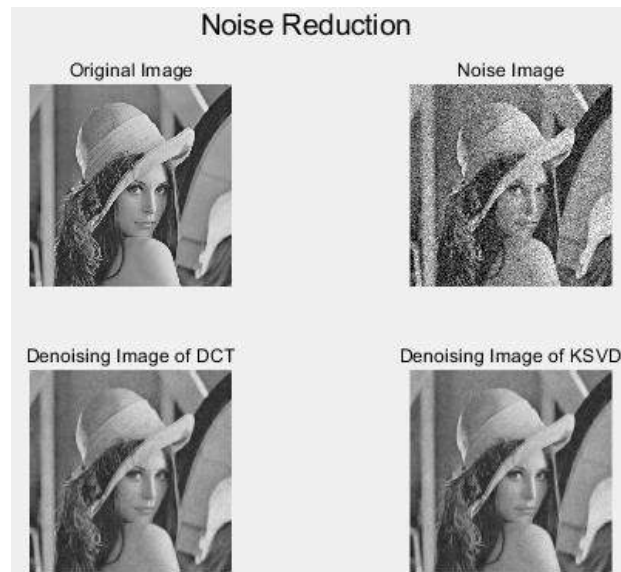
### 4.2.4 DCT 与 K-SVD 去噪性能分析

核心代码: 代码基本上和上面相同, 此处不再陈述。

**说明及数值结果:** 由图像重构可以得到去噪图像。其中, OMP 算法停止条件  $T$  同样取  $(C\sigma)^6$ , 即当  $\|\mathbf{D}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 \leq (C\sigma)^6$  时, 停止。其中  $C=1.15$ 。去噪性能将用 PSNR 来衡量, 其定义为:

$$\text{PSNR} = 10 \log_{10} \left( \frac{M \times N}{\sum_{i,j} (s_{i,j} - \hat{s}_{i,j})^2} \right)$$

去噪后重构图像结果为:



其中，噪声图像的 PSNR 为：

```
PSNR_noise =  
  
20.2426
```

DCT 去噪图像的 PSNR 为：

```
PSNR_DCT =  
  
28.2314
```

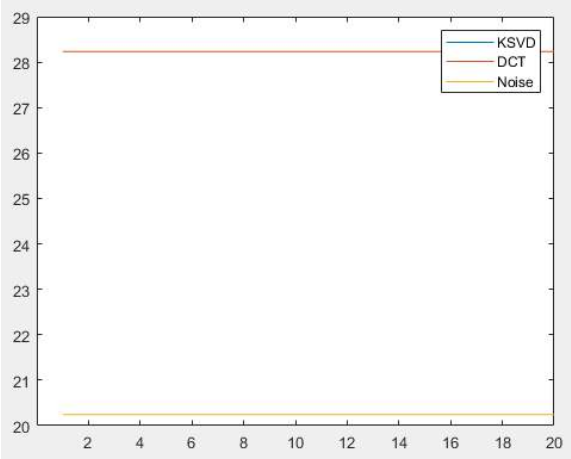
KSVD 去噪图像的 PSNR 为：

J	1	2	3	4	5	6	7
PSNR	28.2323	28.2326	28.2324	28.2330	28.2326	28.2329	28.2326
J	8	9	10	11	12	13	14
PSNR	28.2321	28.2322	28.2329	28.2319	28.2324	28.2319	28.2324
J	15	16	17	18	19	20	
PSNR	28.2321	28.2333	28.2325	28.2332	28.2326	28.2329	

对其取平均得到：

```
>> mean(PSNR_KSVD)  
  
ans =  
  
28.2325
```

这些去噪性能用曲线描述：



发现，这两种方法的去噪效果几乎相等，都具有很好的去噪性能。并且此时用来稀疏表示的平均原子个数为：

Atom_meanDCT	0.4959
Atom_meanKSVD	0.4957

即各仅用了  $\frac{0.4959}{8 \times 8} = 0.7748\%$ ， $\frac{0.4957}{8 \times 8} = 0.7745\%$  的数据表达了去噪图像。

## 第5章 总结与感悟

本文首先从求欠定线性方程组的稀疏解问题入手，因为  $l_0$  范数解是一个非凸的 NP-Hard 问题，因此我们通过以下两种算法来求该问题的近似解：贪心算法和凸松弛算法。在第二章详细介绍了典型的贪心算法：MP 算法和 OMP 算法的原理。然后这些内容跟图像去噪问题关联起来，因为图像去噪问题相当于也是一个带约束条件（稀疏性）的最小化问题。虽然最理想图像去噪方法是直接从噪声图像直接减去噪声项恢复出原始图像，可是在实际情况这个准确的噪声项是很难得到的。因此该问题可以通过某种去噪算法来解决。

图像去噪是一个很有挑战性并一直非常活跃的研究领域。特别地，近年来基于稀疏表示的图像去噪算法受到广泛的关注。因此，第三章主要介绍基于字典学习的稀疏表示以及去噪算法。该方法可分为两种，一种是先固定字典为 DCT 过完备字典，再去求最稀疏解的方法，另一种是利用 K-SVD 算法通过同时更新字典和稀疏解，得到具有更好稀疏表示特性的字典。第四章进行了对它们的仿真实验以及分析。发现这两种方法都具有很好的去噪性能。

## 参考文献

- [1] Aharon M . An algorithm for designing overcomplete dictionaries for sparse representation[J]. IEEE Transactions on Signal Processing, 2006, 54.
- [2] A. Buades, B. Coll, J. M. Morel, “A review of image denoising algorithms, with a new one,” *SIAM J. Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490-530, 2005.
- [3] M. Elad, M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Processing*, vol. 15, no. 12, pp. 3736-3745, 2006.
- [4] M. Elad, *et al*, “On the role of sparse and redundant representations in image processing,” *Proc. IEEE*, vol. 98, no. 6, pp. 972-982, June 2010.
- [5] A. Bruckstein, D. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM Review*, vol. 51, pp. 34-81, 2009.
- [6] 张贤达. 矩阵分析与应用[M]. 北京：清华大学出版社，2004.
- [7] Y. C. Pati, R. Rezaeiifar and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, 1993, pp. 40-44 vol.1. doi: 10.1109/ACSSC.1993.342465
- [8] Donoho D L . Compressed sensing[J]. IEEE Transactions on Information Theory, 2006, 52(4):1289-1306.