利用 FFT 实现基于 MP 的信号稀疏分解

尹忠科[①] 邵 君[①] Pierre Vandergheynst[②]
[①](西南交通大学计算机与通信工程学院 成都 610031)
[②](信号处理实验室，瑞士联邦高工 (洛桑) 1015 瑞士)

摘 要 该文研究基于 Matching Pursuit (MP)方法实现的信号稀疏分解算法，通过对信号稀疏分解中使用的过完备原子库结构特性的分析，提出了一种新的信号稀疏分解算法。该算法首先通过利用原子库的结构特性，很好地处理了稀疏分解过程中计算量和存储量之间的关系。在此基础上，把信号稀疏分解中计算量很大的内积运算转换成互相关运算，最后用 FFT 实现互相关运算，从而大大提高了信号稀疏分解的速度。算法的有效性为实验结果所证实。

关键词 信号处理，稀疏表示，稀疏分解，Matching Pursuit(MP)，FFT
中图分类号：TN911.72 **文献标识码**：A **文章编号**：1009-5896(2006)04-0614-05

MP Based Signal Sparse Decomposition with FFT

Yin Zhong-ke[①] Shao Jun[①] Pierre Vandergheynst[②]
[①](*School of Comp. & Comm. Engineering, Southwest Jiaotong University, Chengdu* 610031, *China*)
[②](*Signal Processing Lab., Swiss Federal Institute of Technology at Laussane* 1015, *Switzerland*)

Abstract In this paper, after study of Matching Pursuit (MP) based signal sparse decomposition, a new sparse decomposition algorithm is presented based on analysis of structure property of the over-complete atom dictionary used in signal sparse decomposition. By making use of the structure property, firstly this new algorithm balances very well computer's speed and memory. Then this algorithm converts very time-consuming inner product calculations in sparse decomposition into crosscorrelation calculations that are fast done by FFT. Therefore the new algorithm improves a lot the speed of signal sparse decomposition. Finally the experimental results show that the performance of the proposed algorithm is very good.

Key words Signal processing, Sparse representation, Sparse decomposition, Matching Pursuit (MP), FFT

1 引言

Mallat 和Zhang首先提出信号在过完备库(over-complete dictionary)上分解的思想[1]。通过信号在过完备库上的分解，用来表示信号的基可以自适应地根据信号本身的特点灵活选取。分解的结果，将可以得到信号一个非常简洁的表达(即：稀疏表示Sparse representation)。而得到信号稀疏表示的过程称为信号的稀疏分解(Sparse decomposition)。由于信号的稀疏表示的优良特性，信号稀疏表示已经被应用到信号处理的许多方面，如信号去噪[1]、信号编码[2]和识别[3]等。其中，在信号时频分布研究方面的应用特别值得关注[4]。但是目前信号的稀疏分解在信号处理中的实际应用很难被推广而产业化。阻碍信号稀疏分解研究及应用发展的关键因素是信号稀疏分解的计算量十分巨大，计算时间在现有计算条件下令人无法忍受。国内有研究人员指出，信号长度为 1024 采样点时，信号的稀疏分解的难度将十分巨大[4]。针对此问题，本文研究基于MP的信号稀疏分解，通过分析稀疏分解中使用的过完备原子库的结构特性，并采用FFT算法，以解决信号稀疏分解计算量大这一难题。

2004-09-09 收到, 2005-02-07 改回
教育部留学回国人员科研启动基金资助课题

2 基于 MP 的信号稀疏分解

2.1 基本思想

基于 MP 的稀疏分解，算法易于理解，是目前信号稀疏分解的最常用方法。假设研究的信号为 f，信号长度为 N。若将信号分解在一组完备正交的基上，则这组基的数目应为 N。由于基的正交性，因而基在由信号所组成的空间中的分布是稀疏的，从而，信号的能量在分解以后将分散分布在不同的基上。这种能量分布的分散最后将导致用基的组合表示信号时表达的不简洁性，即信号表示不是稀疏的。非稀疏的表示，不利于信号的处理，如识别和压缩等。为了得到信号

的稀疏表示，基的构造必须使得基在信号组成的空间中足够的密。由此，基的正交性将不再被保证，所以此时的基也不再是真正意义上的基了，而改称为原子。由这些原子组成的集合，是过完备的，被称为过完备库(over-complete dictionary of atoms)。信号在过完备库上的分解结果一定是稀疏的[1]。

设 $D=\{g_\gamma\}_{\gamma\in\Gamma}$ 为用于进行信号稀疏分解的过完备库，g_γ 为由参数组 γ 定义的原子。用不同的方法构造原子，参数组 γ 所含有的参数及参数个数也不一样。原子 g_γ 的长度与信号本身长度相同，但原子应作归一化处理，即 $\|g_\gamma\|=1$。Γ 为参数组 γ 的集合。由库的过完备性可知，参数组 γ 的个数应远远大于信号的长度，即若用 P 表示过完备库 $D=\{g_\gamma\}_{\gamma\in\Gamma}$ 中原子的个数，则 P 应远远大于信号长度 N。

MP方法分解信号过程如下[1]：

首先从过完备库中选出与待分解信号最为匹配的原子 g_{γ_0}，它满足以下条件：

$$\left|\left\langle f,g_{\gamma_0}\right\rangle\right|=\sup_{\gamma\in\Gamma}\left|\left\langle f,g_\gamma\right\rangle\right| \tag{1}$$

因此信号可以分解为在最佳原子 g_{γ_0} 上的分量和残余两部分，即为

$$f=\left\langle f,g_{\gamma_0}\right\rangle g_{\gamma_0}+R^1f \tag{2}$$

其中 R^1f 是用最佳原子对原信号进行最佳匹配后的残余。对最佳匹配后的残余可以不断进行上面同样的分解过程，即

$$R^kf=\left\langle R^kf,g_{r_k}\right\rangle g_{\gamma_k}+R^{k+1}f \tag{3}$$

其中 g_{γ_k} 满足：

$$\left|\left\langle R^kf,g_{\gamma_k}\right\rangle\right|=\sup_{\gamma\in\Gamma}\left|\left\langle R^kf,g_\gamma\right\rangle\right| \tag{4}$$

由式(2)和式(3)可知，经过 n 步分解后，信号被分解为

$$f=\sum_{k=0}^{n-1}\left\langle R^kf,g_{\gamma_k}\right\rangle g_{\gamma_k}+R^nf \tag{5}$$

其中 R^nf 为原信号分解为 n 个原子的线性组合后，用这样的线性组合表示信号所产生的误差。由于每一步分解中，所选取的最佳原子满足式(4)，所以分解的残余 R^nf 随着分解的进行，迅速地减小。已经证明[1]，在信号满足长度有限的条件下(对数字信号而言，这是完全可以而且一定满足的)，$\|R^nf\|$ 随 n 的增大而指数衰减为 0。从而信号可以分解为

$$f=\sum_{k=0}^{\infty}\left\langle R^kf,g_{\gamma_k}\right\rangle g_{\gamma_k} \tag{6}$$

事实上，由于 $\|R^nf\|$ 的衰减特性，一般而论，用少数的原子(与信号长度相比较而言)就可以表示信号的主要成分，即

$$f\approx\sum_{k=0}^{n-1}\left\langle R^kf,g_{\gamma_k}\right\rangle g_{\gamma_k} \tag{7}$$

其中 $n<<N$。式(7)和条件 $n<<N$ 集中体现了稀疏表示的思想。

虽然基于 MP 的稀疏分解是目前信号稀疏分解的最常用方法，也是几乎所有算法中速度最快的，但和其它的信号稀疏分解方法一样，其存在的关键问题仍是计算量十分巨大。在基于 MP 的信号稀疏分解中，每一步都要完成信号或信号分解的残余在过完备库中的每一个原子上的投影计算。按式(4)所要求，每一步分解实际上要进行的内积计算 $\langle R^kf,g_\gamma\rangle$ 是一个在很高维(N 维)空间的内积计算，而且要进行很多次(P 次)，这是 MP 信号稀疏分解计算量巨大的根本原因所在。

2.2　算法流程

很多高校和科研机构(如：麻省理工学院、纽约大学、瑞士联邦高工和 HP 等)对基于 MP 的信号稀疏分解的具体算法进行了大量的研究，但可惜的是，他们很少发表他们的研究结果，而是把提出的具体算法申请专利。这也是提出信号稀疏分解算法的 Mallat 开创的先例，Mallat 提出信号稀疏分解的概念后，就对一种快速算法申请了专利。通过作者国外的学习和研究，了解到基于 MP 的信号稀疏分解的具体算法大致可分为两类。在图 1 中分别给出了两类算法的粗略的算法流程图。

从算法流程图我们可以看出，两者本质的区别在于过完备原子库的形成方法上。其中第 1 类算法，要求在进行稀疏分解之前，先形成用于进行信号稀疏分解的过完备原子库，在整个分解的过程中，程序使用同一个已经生成的过完备原子库，我们可以形象地称其为“一次生成，终身使用”。在第 2 类方法中，不要求在稀疏分解之前形成原子库。实质上，在第 2 类方法中，从来没有形成一个完整的过完备原子库。它一边生成原子，一边进行最佳原子的搜索，一边删除生成的原子。

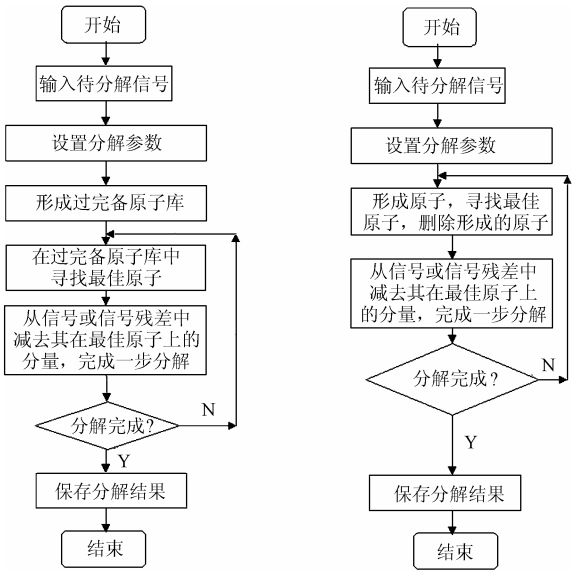

图 1　基于 MP 的信号稀疏分解算法流程图

为什么两类方法有如此大的差别呢？关键在于两类方法是针对不同的计算条件设计的。为了说明基于 MP 的信号

稀疏分解所要求的计算条件，必须对信号稀疏分解中所要生成的过完备原子库有一个明确的概念。前面 2.1 节中稀疏分解的概念说明，得到信号的稀疏分解的前提条件是形成一个过完备的库，即若用 P 表示过完备库中原子的个数，则 P 应远远大于信号长度 N 。过完备库中原子的个数 P 的值，是非常非常大的。这使得第 1 类算法，为了“一次生成，终身使用”原子库，必须要使用巨大的计算机内存。即第 1 类算法相对于第 2 类算法，对计算机的要求是，计算机必须有巨大的内存。第 2 类算法实际上是针对在计算机内存满足不了要求而又必须进行稀疏分解的情况设计的。由于没有生成过完备的原子库，而要进行稀疏分解又必须用过完备原子库，所以，第 2 类算法在进行分解的每一步中，一边生成原子，一边比较搜索最佳的原子，一边还要删除生成的原子。所以第 2 类算法对计算机的内存没有太高的要求。但从上面的分析可知，第 2 类算法每分解一步，实际上都产生了过完备原子库，只是没有保存而已，所以第 2 类算法的计算量是十分惊人的。尽管第 1 类算法的计算量也是很巨大的，但第 2 类算法的计算量相对第 1 类算法而言，还要大得多。从以上分析可知，从计算理论上讲，两者的主要区别在于对计算量和存储量之间关系的不同处理上。

从上面分析可知，第 1 类算法使用了巨大的内存，第 2 类算法几乎对内存没有特别的要求，但计算速度很慢。两类算法在内存的使用上极不平衡，这是很不合理的。下面我们分析稀疏分解中过完备原子库的结构特性，从而设计出一般计算机就能满足内存要求、计算速度又较快的基于 MP 的信号稀疏分解算法。

3 稀疏分解中过完备原子库的结构特性

为了分析稀疏分解中过完备原子库的结构特性，我们必须给出一个具体的过完备原子库。不失一般性，我们引用文献[3]中过完备原子库的形成方法。以下一段内容完全引自文献[3]。过完备原子库由 Gabor 原子组成。一个 Gabor 原子由一个经过调制的高斯窗函数组成：

$$g_\gamma(t)=\frac{1}{\sqrt{s}}g\left(\frac{t-u}{s}\right)\cos(vt+w) \tag{8}$$

其中 $g(t)=e^{-\pi t^2}$ 是高斯窗函数，$\gamma=(s,u,v,w)$ 是时频参数。时频参数可以按以下方法离散化：$\gamma=(a^j,pa^j\Delta u,ka^{-j}\Delta v,i\Delta w)$，其中，$a=2$，$\Delta u=1/2$，$\Delta v=\pi$，$\Delta w=\pi/6$，$0<j\le\log_2 N$，$0\le p\le N2^{-j+1}$，$0\le k<2^{j+1}$，$0\le i\le 12$。上面的描述就给出了一个具体的过完备原子库。

通过对上面参数的分析计算，我们可以知道，如果信号长度 N 为 256，则过完备原子库中原子的个数为 126490 个。对一般的计算机内存而言，这样的数据量太大了，根本无法存储这样的一个原子库。只有高档的计算机才能有此存储能力。所以基于 MP 的信号稀疏分解的第 1 类算法只有高档的计算机才能实现。第 2 类算法用一般的计算机进行稀疏分解，由于无法存储如此巨大的原子库，只有在分解每一步时，临时生成 126490 个原子以便选取最佳的原子，所以速度十分缓慢。

在过完备原子库中，一个原子由 4 个参数 (s,u,v,w) 决定，其中 s 是伸缩因子(尺度因子)，u 是原子的平移因子(位移因子)，v 是原子的频率，w 是原子的相位。在稀疏分解理论中，原子库中原子的个数是一个重要的因素，但更为重要的一个因素是过完备原子库必须有好的结构。(可惜的是，后者的研究几乎没有见到报道)。而原子库的结构就是由以上 4 个参数决定的。为了评价一个原子库的结构的好坏，我们必须定义原子库具有良好结构的含义，我们认为，一个原子库具有好的结构，包含两个方面的含义：一是原子库中应包含尽可能多的原子个数和种类，以达到稀疏分解的目的，获得稀疏分解的良好效果；二是原子库中应尽可能不包含相近似的原子，以满足存储量和计算量方面的要求。以上两个方面是相互矛盾的，前者更侧重于理论，后者更侧重于实际计算。当二者达到一个非常好的平衡时，原子库的结构将是最佳的。

通过研究，我们发现，如果按照上面原子库良好结构的定义，现在已经发表的原子库结构都是不太合理的。以上面的由文献[3]给出的原子库为例，我们说明这种结构的不合理性。参数 s，v 和 w 定义了一个原子的形状，而参数 u 定义了一个原子的中心位置。如果原子库中的不同原子，具有相同的参数 s，v 和 w，不同的参数 u，这样的原子各方面的特征都相同，只有中心点位置不同而已。从纯理论上讲，这种不同的原子包含在原子库中是完全必要的，但从实际计算的角度讲，原子库中包含众多这样的原子是根本没有必要的。它违背了上述原子库良好结构定义中的第 2 个含义。因此，一个具有良好结构的原子库中，参数 u 应是不变的，且 $u=N/2$ 。

4 利用 FFT 实现基于 MP 的信号稀疏分解

利用信号稀疏分解中的过完备库的结构特性和 FFT 的快速计算的特性，我们提出了以下 3 个算法，这 3 个算法相互关联，后面的算法以前面的算法为基础，且后面算法较前面算法速度一步步提高。

4.1 算法 1

根据对具有良好结构的原子库的分析，可以构造出基于

MP 的信号稀疏分解的快速算法。快速算法的基本思想如下：(1)原子库中，原子的参数 s，v 和 w 的选取应和其它方法一样，参数 u 应是不变的，且 $u=N/2$。这样原子库的大小将大大减小，适合一般计算机内存的要求，克服了第 1 类算法的缺点；(2)MP 稀疏分解过程：在分解的每一步，对于参数为 s_i，v_i，w_i 和 u_i 的原子($u_i \neq N/2$)，从原子库中取出参数为 s_i，v_i，w_i 和 $u=N/2$ 的原子，通过平移，即可得到参数为 s_i，v_i，w_i 和 u_i 的原子($u_i \neq N/2$)。由于原子的平移几乎不需要计算量，所以，整个稀疏分解过程的速度和第 1 类方法几乎一样。从上面的描述可以看出，我们提出的算法，达到了计算量和存储量之间的一个很好的平衡。

4.2　算法 2

上面的算法 1 大大提高了基于 MP 的信号稀疏分解第 2 类算法的速度，而稀疏分解的效果保持不变。沿着上面的思路继续思考下去，我们提出了一种新的算法，该算法不但提高了信号稀疏分解的速度，而且同时提高了信号稀疏分解的效果。

算法 2 中原子库的形成方法和算法 1 中一样，因此存储量能够适合一般计算机的性能。在稀疏分解的过程中，对于原子库中的一个原子(参数：s_i，v_i，w_i 和 $u=N/2$)，让 u_i 取所有可能的值 $[0,N-1]$。这样相当于原子库的大小增加了，因此信号稀疏分解的效果要好一些。这样做将增加计算量，但通过下面的方法，增加的计算量不会影响总体的计算速度的提高。在 2.1 节中我们已经讨论过，在原子库中寻找最佳原子的过程中，主要的计算量是花费在计算内积 $\langle R^k f, g_\gamma \rangle$ 上。对于具有参数 s_i，v_i，和 w_i 的原子库中的一个原子 g_γ，如果要让 u_i 取所有可能的值 $[0,N-1]$，则该原子要和信号或信号的残差作 N 次内积 $\langle R^k f, g_\gamma \rangle$ 计算。由于 u_i 连续取值，且从 0 到 $N-1$，所有 N 次内积 $\langle R^k f, g_\gamma \rangle$ 计算可以转换成一次 $R^k f$ 和 g_γ 的互相关运算 $r_{R^k f, g_\gamma}$。虽然从理论上讲，这样的转换，计算量几乎没有变化，但是计算效率却大大提高。

4.3　算法 3

算法 3 是利用 FFT 算法对算法 2 的进一步改进。在算法 2 中，我们已经把基于 MP 的信号稀疏分解中花费绝大部分计算时间的内积 $\langle R^k f, g_\gamma \rangle$ 运算，转变成了互相关运算 $r_{R^k f, g_\gamma}$，其中 N 次内积 $\langle R^k f, g_\gamma \rangle$ 计算可以转换成一次 $R^k f$ 和 g_γ 的互相关运算 $r_{R^k f, g_\gamma}$。由于利用 FFT 算法可以快速实现互相关运算，所以我们利用 FFT 算法对算法 2 进行进一步的改进。这样的改进，丝毫不会影响信号稀疏分解的效果，但却可以大大提高稀疏分解的速度。因为一般而言，用 FFT 算法实现互相关运算，可以使计算速度提高 1~2 个数量级。

5　实验结果与分析

实验中采用长度为 256 取值范围为[0, 255]的实际信号。过完备库中原子的构造方法按文献[3]。当信号长度为 256 时，根据此文献，库的大小为含有 126490 个原子，以满足库的过完备性。由于信号稀疏分解的速度依赖于计算条件(硬件条件和软件条件)，所以给出实际的稀疏分解时间是没有太大的参考意义的(几乎没有文献给出信号稀疏分解的计算时间)。我们把基于 MP 的信号稀疏分解第 2 类算法的速度设为 1，在表 1 中给出了本文中的算法的速度是基于 MP 信号稀疏分解第 2 类算法的速度的多少倍。由于基于 MP 的信号稀疏分解的第 1 类算法需要的存储量远远超出我们实验用的计算机内存，所以我们没有做关于第 1 类算法的实验。

从经典 FFT 理论上分析，算法 3 应比算法 2 速度快至少 1~2 个数量级，从而算法 3 比基于 MP 的信号稀疏分解第 2 类算法速度快达数百倍以上。但我们实际的实验结果表明并不是这样。这可能由于我们使用的编程语言是 Matlab 使得 FFT 算法的特性没有很好的体现出来，或者是由于我们程序中仍然存在有待改进的地方。我们正在对此问题展开分析和研究。

图 2 中给出了实际的信号及利用不同方法稀疏分解为 30 个原子后重建的信号。可以看出，重建信号较好地近似表示了原信号。几种分解方法重建的信号从直观上讲相差无几，但如果用重建信号与原始信号的均方误差来衡量的话，基于 MP 的信号稀疏分解(第 2 类算法)和算法 1，两者重建的信号是完全一样的，而算法 2 和算法 3 重建的信号完全相同，且

表 1　信号稀疏分解相对速度比较

算法	计算速度(倍)
基于 MP 的信号稀疏分解(第 2 类算法)	1
算法 1	5.2
算法 2	7.9
算法 3	29.5

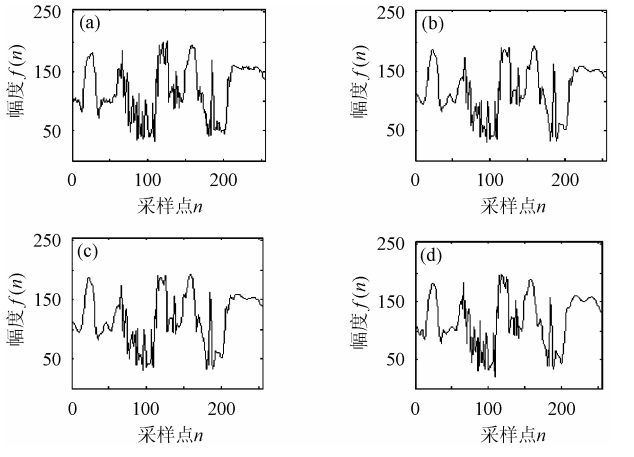

图 2　信号和稀疏分解后重建的信号
(a)原始信号　(b)第 2 类算法重建信号
(c)算法 1 重建信号　(d)算法 2 重建信号

重建的信号更好，即更加接近于原始信号。(由于算法 2 和算法 3 重建的信号完全一样，图中没有给出算法 3 重建的信号)。

实验中用 30 个原子即可表示原始信号，也足以说明这种表示的稀疏性。同时，用 30 个原子即可表示原信号主要特征，表明这种稀疏表示为进行信号处理(如压缩、识别等)提供了极其有利的条件。

参 考 文 献

[1] Mallat S, Zhang Z. Matching pursuit with time-frequency dictionaries[J]. *IEEE Trans. on Signal Processing*, 1993, 41(12): 3397 – 3415.

[2] 张文耀. 基于匹配跟踪的低位率语音编码研究[D]. [博士论文]，中国科学院研究生院(软件研究所). 2002 年 10 月.

[3] Arthur P L, Philipos C L. Voiced/unvoiced speech discrimination in noise using Gabor atomic decomposition[A]. Proc. Of IEEE ICASSP[C], Hong Kong, April, 2003, VoL I: 820 – 828.

[4] 邹红星，周小波，李衍达. 时频分析：回溯与前瞻[J]. 电子学报，2000，28(9)：78 – 84.

尹忠科： 男，1969 年生，博士后，副教授，研究方向为信号与信息处理、图像传输与处理.

邵 君： 女，1982 年生，硕士，研究方向为信号与信息处理；

Pierre Vandergheynst： 男，1972 年生，比利时人，博士，教授，研究方向为信号与信息处理、图像传输与处理.