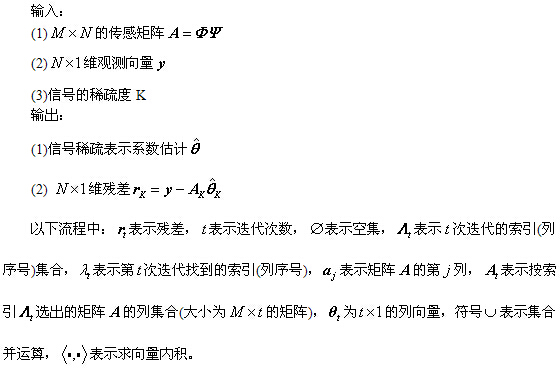
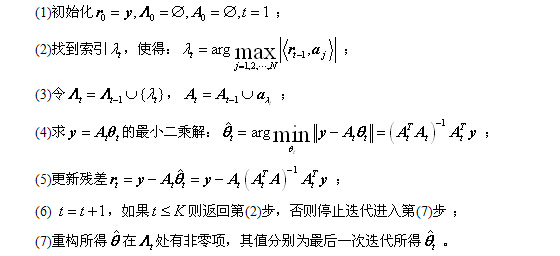
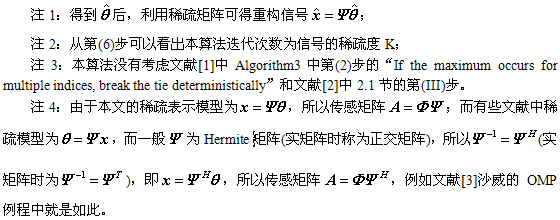
OMP重构算法流程：







2、正交匹配追踪(OMP)MATLAB代码(CS\_OMP.m)

function [ theta ] = CS\_OMP( y,A,t )

%CS\_OMP Summary of this function goes here

%Version: 1.0 written by jbb0523 @2015-04-18

% Detailed explanation goes here

% y = Phi \* x

% x = Psi \* theta

% y = Phi\*Psi \* theta

% 令 A = Phi\*Psi, 则y=A\*theta

% 现在已知y和A，求theta

[y\_rows,y\_columns] = size(y);

if y\_rows<y\_columns

y = y';%y should be a column vector

end

[M,N] = size(A);%传感矩阵A为M\*N矩阵

theta = zeros(N,1);%用来存储恢复的theta(列向量)

At = zeros(M,t);%用来迭代过程中存储A被选择的列

Pos\_theta = zeros(1,t);%用来迭代过程中存储A被选择的列序号

r\_n = y;%初始化残差(residual)为y

for ii=1:t%迭代t次，t为输入参数

product = A'\*r\_n;%传感矩阵A各列与残差的内积

[val,pos] = max(abs(product));%找到最大内积绝对值，即与残差最相关的列

At(:,ii) = A(:,pos);%存储这一列

Pos\_theta(ii) = pos;%存储这一列的序号

A(:,pos) = zeros(M,1);%清零A的这一列，其实此行可以不要，因为它与残差正交

%y=At(:,1:ii)\*theta，以下求theta的最小二乘解(Least Square)

theta\_ls = (At(:,1:ii)'\*At(:,1:ii))^(-1)\*At(:,1:ii)'\*y;%最小二乘解

%At(:,1:ii)\*theta\_ls是y在At(:,1:ii)列空间上的正交投影

r\_n = y - At(:,1:ii)\*theta\_ls;%更新残差

end

theta(Pos\_theta)=theta\_ls;%恢复出的theta

end

function [A]=OMPerr(D,X,errorGoal);

%=============================================

% Sparse coding of a group of signals based on a given

% dictionary and specified number of atoms to use.

% input arguments: D - the dictionary

% X - the signals to represent

% errorGoal - the maximal allowed representation error for

% each siganl.

% output arguments: A - sparse coefficient matrix.

%=============================================

[n,P]=size(X);

% X应该是64X62001

[n,K]=size(D);

% 初始D只是64X256

E2 = errorGoal^2\*n;

maxNumCoef = n/2;

% A为256X62001

A = sparse(size(D,2),size(X,2));

% P=62001

for k=1:1:P,

a=[];

x=X(:,k);

%先取原图一列 这个是原图一列

residual=x;

indx = [];

a = [];

%原图每列的平方

currResNorm2 = sum(residual.^2);

j = 0;

while currResNorm2>E2 & j < maxNumCoef,

j = j+1;

proj=D'\*residual;

%proj表示字典与原图每一列相乘

pos=find(abs(proj)==max(abs(proj)));

pos=pos(1);

%将得到与原图乘出最大的行数拿出来

indx(j)=pos;

a=pinv(D(:,indx(1:j)))\*x;

residual=x-D(:,indx(1:j))\*a;

currResNorm2 = sum(residual.^2);

end;

if (length(indx)>0)

A(indx,k)=a;

end

end;

return;